

Path Planning for Line Marking Robots Using 2D Dubins' Path

Ibrahim A. Hameed^(✉)

Faculty of Engineering and Natural Sciences,
Department of Automation Engineering (AIR),
Norwegian University of Science and Technology,
NTNU in Ålesund, Postboks 1517, 6025 Ålesund, Norway
ibib@ntnu.no

Abstract. This paper proposes a path planning algorithm based on 2D *Dubins' path* for the construction of a curvature continuous trajectory for the autonomous guidance of line marking robots in football stadiums. The algorithm starts with four corner points representing the playable football field and generates a set of waypoints representing various parts of the field layout such as touch and goal lines, goal and penalty area, center line and mark, corner and penalty arcs, center mark and center circle, and penalty marks. A complete, continuous and smooth path is then generated by connecting these waypoints using 2D Dubins' path in a way to ensure that the generated path takes into account the dynamic constraints of the vehicle (such as maximum curvature and velocity), keep the vehicle at a safe distance from obstacles, and not harm the field grass. The efficiency of the algorithm is tested using simulation and in reality. Results showed that the algorithm is able to reliably plan a safe path in real time able to command the line marking robot with high accuracy and without the need for human guidance. The path planning algorithm developed in this paper is implemented in both Matlab and Python.

Keywords: Guidance and navigation · Robotics in sports · FIFA · Line marking and illustration · Dubins' curve · Football

1 Introduction

The central idea of most robotics applications is to eliminate the need for a human operator. The most obvious reason is; (1) to save labour and reduce cost where robots can conduct certain tasks far more rapidly and with greater precision than can human, (2) the human is bad for the production process such as in semiconductor and food handling, and pharmaceuticals where the use of robots reduce contamination, and (3) the product is bad for the human such as handling hazardous materials in chemical and radioactive plants where the use

of robots minimise risks to human operators. Another reason is that in aging societies that many countries are facing, the need arises to allocate work to smart machines, as in the case of industrial robots [1].

To date, robots have been very successful at manipulation in controlled environments such as in factories and in laboratories. Historical attempts of producing robotic arms went back to mid 50s when the first programmable robot arm for high speed handling was designed by George Devol in 1954 [2]. The use of robotics in laboratory, pharmaceutical industry and environmental monitoring went back to late 80s [3]. The use of robotic has been extended, since then, to include domains such as industrial maintenance and repair [4], outdoor robots for domains such as agriculture [5–7], animal-farming [8], mining [9], construction [10,11], power plants [12], oil and gas industry [13], space exploration [14], security and defense [15,16], etc.

Path-planning systems are of great significance when it comes to the performance and mission accomplishment of practically every type of outdoor robotics. Determination of a collision free path that a robot can follow between start and goal positions through obstacles cluttered in a workspace is central to the design of each autonomous robot to accomplish tasks without or with minimal human guidance [17]. Coverage Path Planning, on the other hand, is the task of determining a path that passes over all points of an area or volume of interest while avoiding obstacles. This task is integral to many robotic applications, such as vacuum cleaning robots, painter robots, autonomous underwater vehicles, demining robots, lawn mowers, automated harvesters, window cleaners, etc. [18,19]. An optimised path for a robot to navigate in an environment is the one which takes into account both the physical constraints of the vehicle, and the workspace constraints, such as obstacles or environmental forces [20,21].

Hameed et al. (2011) presented a promising application of field robotics for automating the periodical operations frequently carried out in football stadiums such as lawn cutting, lawn stripping and line marking in football stadiums [22]. The manual operation of these tasks is very expensive, very frequent, and requires very skilled and well trained personnel. In an attempt to reduce the cost, line marking or field panting is done less frequently through the use of long lasting paints. These types of paints contains high level of metal acetate salts which can last longer, however, it is more expensive and can kill the turf. A proposed solution is to use low cost paints which cannot harm the grass and painting more frequent. This solution can be feasible if this task can be performed autonomously using low weight vehicle able to work around the clock with high precision and with minimum human intervention. The line marking robot is a small three or four wheeled vehicle equipped with a line marking spray nozzle able to provide homogeneously painted lines and curves of constant width (i.e., 12 cm width). The proposed system is expected to reduce the amount of paint needed per each football field to an absolute minimum and hence reduce cost and possible environmental impact.

In this paper, a complete curvature and continuous path is constructed by connecting the field layout components such as straight lines, arcs and circles using Dubins' paths. The generated trajectory takes into account both the

physical constraints of the vehicle such as the minimum turning radius of the vehicle and the workspace constraints such as obstacles. The remainder of the paper is organized as follows: standard FIFA football field layout and dimension are presented in Sect. 2. In Sect. 3, the algorithm for generating waypoints representing field layout is presented. Dubins' path and its use in connecting layout components to construct a continuous, smooth, and safe trajectory for robot navigation are presented in Sect. 4. In Sect. 5, the ability of the developed algorithm in generating a complete and smooth trajectory for a simulated field is presented. Finally, concluding remarks and future work are presented in Sect. 6.

2 Standard FIFA's Football Field Dimensions and Layout

The Fédération Internationale de Football Association (FIFA) (i.e., International Football Association in English) is an association governed by Swiss law founded in 1904 and based in Zurich. It has 209 member associations and its goal is the constant improvement of football game around the world. The rule of FIFA is to define and amend the Laws of the game on behalf of the global football community for the global game to grow and thrive. The official dimensions of football fields for international matches are specified in Law 1 of the official FIFA Laws of the Game, known as the field of play law. According to the FIFA rules for international matches, dimensions and parts of the field of play are defined as follows [23]:

2.1 Field Markings

The field of play must be rectangular and marked with lines. These lines belong to the areas of which they are boundaries. The two longer boundary lines are called touch lines. The two shorter lines are called goal lines. The field of play is divided into two halves by a halfway line, which joins the midpoints of the two touch lines. The centre mark is indicated at the midpoint of the halfway line. A circle with a radius of 9.15 m is marked around it, as it is shown in Fig. 1a.

2.2 Field Dimensions

The length of the touch line must be greater than the length of the goal line. The two touch lines must be of the same length, and be between 90 and 120 m in length (or between 100 and 110 m for international matches). The two goal lines must be of the same length, and be between 45 and 90 m (or between 64 and 75 m for international matches). All lines must be equally wide, not to exceed 12 cm, as it is shown in Fig. 1b.

2.3 Goal Area

Two lines are drawn at right angles to the goal line, 5.5 m from the inside of each goalpost. These lines extend into the field of play for a distance of 5.5 m and are joined by a line drawn parallel with the goal line. The area bounded by these lines and the goal line is the goal area.

2.4 Penalty Area

Two lines are drawn at right angles to the goal line, 16.5 m from the inside of each goalpost. These lines extend into the field of play for a distance of 16.5 m and are joined by a line drawn parallel with the goal line. The area bounded by these lines and the goal line is the penalty area. Within each penalty area, a penalty mark is made 11 m from the midpoint between the goalposts and equidistant to them. An arc of a circle with a radius of 9.15 m from the centre of each penalty mark is drawn outside the penalty area.

2.5 Corner Arcs

A flagpost is placed at each corner of the field. A quarter circle with a radius of 1 m (from each corner flagpost) is drawn inside the field of play).

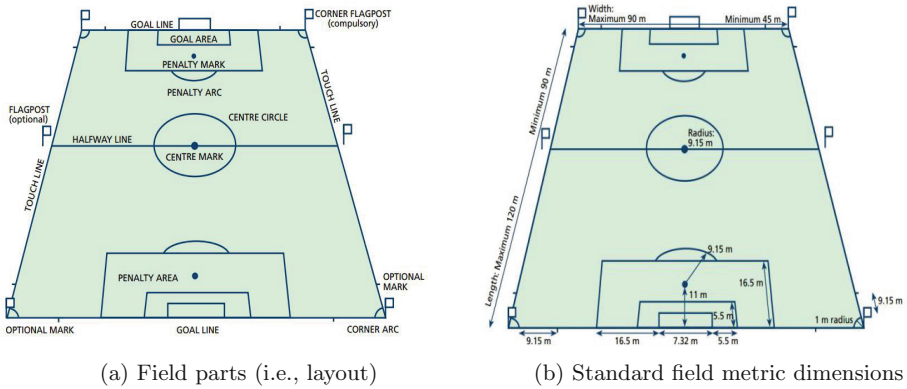


Fig. 1. Standard FIFA field of play dimensions and layout

3 Waypoint Generation of the Field Layout

In this section, waypoints representing the field of play parts are generated. The input to the algorithm are four corner points of the field of play and the outputs are the coordinates of all field parts.

3.1 Field of Play Coordinates

The algorithm receives four coordinates of the green rectangle (i.e., playable field), $P_i = (x_i, y_i)$ where $i = 1, 2, 3,$ and 4 . The two lower and upper goal lines are of length, w , and are represented by line segments $\overline{P_1P_2}$ and $\overline{P_3P_4}$ respectively. The two right and left touch lines are of length, l , and are represented by $\overline{P_2P_3}$ and $\overline{P_4P_1}$ respectively. To generate a rectangle field, width and length offset value is calculated over each goal and touch lines using the following two equations, respectively:

$$w_{ij} = |\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} - w|/2 \tag{1}$$

$$l_{ij} = |\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} - l|/2 \tag{2}$$

Two function, based on simple linear algebra, for line representation, $line_{ij} = createLine(p_i, p_j)$, for creating a line segment using its starting and ending points, p_i and p_j , and a function for finding a point p on a line segment at a distance d from its starting point, $p = pointOnLine(line, d)$, are used to find the four corner points of the field of play [24]. The corner points, C_i for $i = 1, 2, 3$, and 4, shown in Fig. 2a, can be generated in the following order:

1. *Represent the playable field (i.e., main rectangle):*
 $line_{12} = createLine(p_1, p_2)$,
 $line_{23} = createLine(p_2, p_3)$,
 $line_{34} = createLine(p_3, p_4)$, and
 $line_{41} = createLine(p_4, p_1)$.
2. *Find intersection points of the field of play lines with the main rectangle:*
 $p_{w_{12}} = pointOnLine(line_{12}, w_{12})$,
 $p_{w_{12}+w} = pointOnLine(line_{12}, w_{12} + w)$,
 $pl_{23} = pointOnLine(line_{23}, l_{23})$,
 $pl_{23+l} = pointOnLine(line_{23}, l_{23} + l)$,
 $p_{w_{34}} = pointOnLine(line_{34}, w_{34})$,
 $p_{w_{34}+w} = pointOnLine(line_{34}, w_{34} + w)$,
 $pl_{41} = pointOnLine(line_{41}, l_{41})$, and
 $pl_{41+l} = pointOnLine(line_{41}, l_{41} + l)$.
3. *Find corner points of the field of play:*
 $C_1 = pointOnLine(createLine(pl_{41+l}, pl_{23}), w_{12})$,
 $C_2 = pointOnLine(createLine(pl_{41+l}, pl_{23}), w_{12} + w)$,
 $C_3 = pointOnLine(createLine(pl_{23+l}, pl_4), w_{12})$, and
 $C_4 = pointOnLine(createLine(pl_{23+l}, pl_4), w_{12} + w)$.

3.2 Centre Line and Centre Mark Coordinates

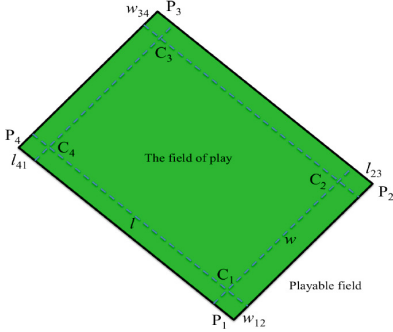
The centre line divides the field into two halves by a halfway line, which joins the midpoints of the two touch lines. The two midpoints can be obtained as follows:

$$MP_{23} = pointOnLine(createLine(C_2, C_3), l/2), \text{ and}$$

$$MP_{41} = pointOnLine(createLine(C_4, C_1), l/2).$$

The centre mark is indicated at the midpoint of the halfway line and is obtained as follows:

$$CM = pointOnLine(createLine(MP_{23}, MP_{41}), w/2).$$



(a) Standard field metric dimensions



(b) Line marking robot (source: <http://www.intelligentmarking.com/> and <http://compleks.com/>)

Fig. 2. Coordinate generation of the field of play and line marking robot

3.3 Goal Area Coordinates

It is the area bounded by two lines drawn at right angles to the goal line, 5.5 m from the inside of each goalpost (i.e., 9.16 m from the midpoint of the goal line) and extended into the field of play for a distance of 5.5 m and are joined by a line drawn parallel to the goal line. The coordinates of the lower and upper goal areas are calculated as follows:

$$\begin{aligned}
 g_{l1} &= \text{pointOnLine}(\text{createLine}(C_1, C_2), w/2 + 9.16), \\
 g_{l4} &= \text{pointOnLine}(\text{createLine}(C_1, C_2), w/2 - 9.16), \\
 g_{u1} &= \text{pointOnLine}(\text{createLine}(C_2, C_3), w/2 + 9.16), \\
 g_{u4} &= \text{pointOnLine}(\text{createLine}(C_2, C_3), w/2 - 9.16), \\
 g_{l2} &= \text{pointOnLine}(\text{createLine}(g_{l1}, g_{u4}), 5.5), \\
 g_{u3} &= \text{pointOnLine}(\text{createLine}(g_{l1}, g_{u4}), l - 5.5), \\
 g_{l3} &= \text{pointOnLine}(\text{createLine}(g_{l4}, g_{u1}), 5.5), \text{ and} \\
 g_{u2} &= \text{pointOnLine}(\text{createLine}(g_{l4}, g_{u1}), l - 5.5).
 \end{aligned}$$

3.4 Penalty Area Coordinates

It is the area bounded by two lines drawn at right angles to the goal line, 16.5 m from the inside of each goalpost (i.e., 20.16 m from the midpoint of the goal line) and extended into the field of play for a distance of 16.5 m and are joined by a line drawn parallel with the goal line. The coordinates of the lower and upper penalty areas are calculated as follows:

$$\begin{aligned}
 p_{l1} &= \text{pointOnLine}(\text{createLine}(C_1, C_2), w/2 + 20.16), \\
 p_{l4} &= \text{pointOnLine}(\text{createLine}(C_1, C_2), w/2 - 20.16), \\
 p_{u1} &= \text{pointOnLine}(\text{createLine}(C_2, C_3), w/2 + 20.16), \\
 p_{u4} &= \text{pointOnLine}(\text{createLine}(C_2, C_3), w/2 - 20.16), \\
 p_{l2} &= \text{pointOnLine}(\text{createLine}(p_{l1}, p_{u4}), 16.5),
 \end{aligned}$$

$p_{u3} = \text{pointOnLine}(\text{createLine}(p_{l1}, p_{u4}), l - 16.5),$
 $p_{l3} = \text{pointOnLine}(\text{createLine}(p_{l4}, p_{u1}), 16.5),$ and
 $p_{u2} = \text{pointOnLine}(\text{createLine}(p_{l4}, p_{u1}), l - 16.5).$

3.5 Penalty Mark Coordinates

A penalty mark is made 11 m from the midpoint between the goalposts and equidistant to them. The coordinates of lower and upper penalty marks can be obtained by finding the midpoints of the lower and upper goal lines (i.e., $gmpl$ and $gmpu$) and then finding its coordinates on the line connecting them as follows:

$gmpl = \text{pointOnLine}(\text{createLine}(C_1, C_2), w/2),$
 $gmpu = \text{pointOnLine}(\text{createLine}(C_3, C_4), w/2),$
 $lpm = \text{pointOnLine}(\text{createLine}(gmpl, gmpu), 11.0),$ and
 $upm = \text{pointOnLine}(\text{createLine}(gmpl, gmpu), l - 11.0).$

3.6 Centre Circle Coordinates

It is a circle with a radius of 9.15 m marked around the centre mark (CM). The coordinates of the centre circle can be obtained for a suitable precision using the parametric equation of a circle which is given by:

$$(x, y) = (x_0, y_0) + r(\cos(\theta), \sin(\theta)) \tag{3}$$

where x and y are the circle coordinates, x_0 and y_0 are the centre mark coordinates, the circle radius $r = 9.15$ m, and the circle angle θ in radians where $0 \leq \theta \leq 2\pi$. In this paper, an angle of a step size of 0.01π is used, bigger values will draw the circle faster but you might notice imperfections, and vice versa.

3.7 Corner and Goal Arcs Coordinates

An arc is a part of the circle with an angle in radians $\theta_{arc} \in [\theta_{start}, \theta_{end}]$ and therefore it can be generated for a suitable angle step size using Eq. 3. θ_{start} and θ_{end} can be easily obtained in terms of the starting and ending point of the arc using the equation:

$$\theta_{start} = \tan^{-1} \frac{y_1 - y_0}{x_1 - x_0} \tag{4}$$

$$\theta_{end} = \tan^{-1} \frac{y_2 - y_0}{x_2 - x_0} \tag{5}$$

where (x_0, y_0) , (x_1, y_1) , and (x_2, y_2) are the arc centre, arc starting, and arc ending points, respectively. Corner arcs, for example, has a radius of 1 m, the starting and ending points of the arc at C_1 can be found as follows:

$ARC_{start}^{C_1} = \text{pointOnLine}(\text{createLine}(C_1, C_2), 1),$ and
 $ARC_{end}^{C_1} = \text{pointOnLine}(\text{createLine}(C_4, C_1), l - 1).$

4 Dubins' Path

Dubins' path in this paper are used to connect the components of the generated field layout such as line segments, arcs and circles in order to generate a continuous, smooth, and complete path which can be used by the robot to navigate throughout the line marking (or field painting) process. Given two points in the plane, the initial and final points, P_i and P_f , respectively. Each point is associated with its own orientation angle, α and β , respectively, which defines the prescribed direction of motion. The combination of (P_i, α) and (P_f, β) , are known as the initial and final configurations. Given (P_i, α) and (P_f, β) , the task is to find the shortest smooth path from P_i to P_f such that it starts and ends with the directions of motions α and β , respectively, and the path curvature is limited by $1/\rho$ where ρ is the minimum turning radius of the vehicle or the robot under consideration.

The first complete solution to this problem was first reported by Dubins in 1957 [25] where he showed that the shortest feasible path consists of exactly three path segments of a sequence CCC or CSC , where C for *circle* is an arc of radius ρ , and S for *straight* is a line segment which can then be decomposed into a set of six candidate paths. The optimum path was then found by explicitly computing all paths on the list and then comparing them which may become a problem in applications where computation time is critical, such as in real-time robot motion planning. Instead, Shkel and Lumelsky presented a logical classification scheme that allows one to extract the shortest path from the Dubins set directly, without explicitly calculating the candidate paths [26]. In this paper, a function for finding the optimal path connecting any initial and final configurations, $path_{P_i, P_f} = \text{Dubins}((P_i, \alpha), (P_f, \beta), \rho)$, is developed.

5 Simulation Results

Assume that a field has the coordinates, starting from lower left corner and in counterclockwise direction, $(-30.7, 45.9)$, $(54.1, 130.7)$, $(110.7, 74.1)$, and $(25.9, -10.7)$ and a robot with a minimum turning radius of 2m with some random heading angle is located at $(-20.4, 54.6)$. The algorithm started by generating the waypoints of the playable field layout and then used Dubins' path to connect the layout components. The generated trajectory for an accuracy of 0.01 m is shown in Fig. 3 and a simulation can be seen on Youtube¹. The complete trajectory can then be used to autonomously guide the line marking robot shown in Fig. 2b throughout the line marking process.

¹ <https://www.youtube.com/watch?v=iKLN0AniZGw>.

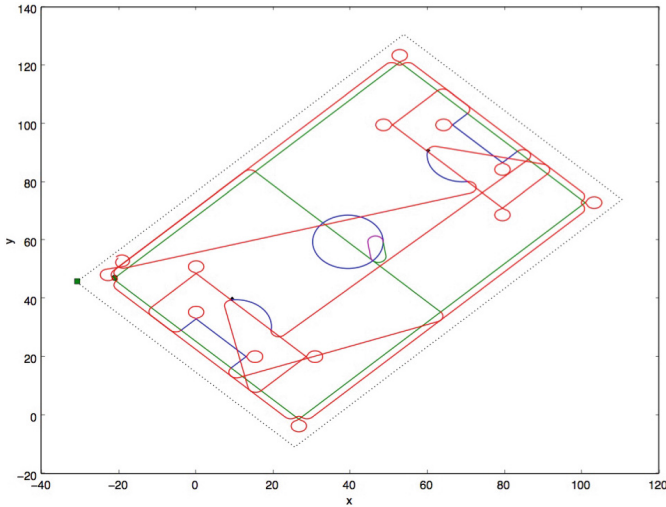


Fig. 3. Coordinate generation of the field of play.

6 Conclusions

In this paper, a path planning algorithm is used to generate the waypoints representing the layout of the field of play according to FIFA's standard dimensions. Dubins' path is used to connect the layout components in a way to generate a complete, continuous, short, and smooth trajectory which can be used safely to navigate the robot throughout the line marking process. Trajectory generation based on clothoid arcs is left for future work.

References

1. Kimitoshi, Y., Ryohei, U., Shunichi, N., Mitsuharu, K., Kei, O., Kiyoshi, M., Masaru, I., Isao, S., Masayuki, I.: Home-assistant robot for an aging society. *Proc. IEEE* **100**(8), 2429–2441 (2012)
2. Michael, E.M.: Evolution of robotic arms. *J. Robot. Surg.* **1**(2), 103–111 (2007)
3. Robert, B.: Robots in the laboratory: a review of applications. *Ind. Robot Int. J.* **39**(2), 113–119 (2012)
4. Parker, L., Draper, J.: Robotics applications in maintenance and repair. In: *Handbook of Industrial Robotics*, 2nd edn. Wiley (1999)
5. van Henten, E.J., Hemming, J., Tuijl, B.V., Kornet, J., Meuleman, J., Bontseman, J., Os, E.V.: An autonomous robot for harvesting cucumbers in greenhouses. *Auton. Robots* **13**(3), 241–258 (2002)
6. van Henten, E.J., Bac, C.W., Hemming, J., Edan, Y.: Robotics in protected cultivation. In: *Proceedings of the 4th IFAC Conference on Modelling and Control in Agriculture, Horticulture and Post Harvest Industry*, IFAC Proceedings Volumes, vol. 46, no. 18, pp. 170–177 (2013)

7. Astrand, B., Baerveldt, A.: An agricultural mobile robot with vision-based perception for mechanical weed control. *Auton. Robots* **13**(1), 21–35 (2002)
8. Andersen, N., Braithwaite, I., Blanke, M., Sorensen, T.: Combining a novel computer vision sensor with a cleaning robot to achieve autonomous pig house cleaning. In: *Proceedings of the 44th IEEE Conference on Decision and Control (CDC)*, Seville, Spain, pp. 8831–8836 (2005)
9. Gary, S., Stentz, A.: A robotic system for underground coal mining. In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 633–638 (1992)
10. Werfel, J., Bar-Yam, Y., Rus, D., Nagpal, R.: Distributed construction by mobile robots with enhanced building blocks. In: *Proceedings of the 2006 IEEE International Conference on Robotics and Automation (ICRA)*, Orlando, FL, pp. 2787–2794 (2006)
11. Zavadskas, E.K.: Automation and robotics in construction: international research and achievements. *Autom. Constr.* **19**(3), 286–290 (2010)
12. Abidi, M., Eason, R., Gonzalez, R.: Autonomous robotic inspection and manipulation using multisensor feedback. *Comput. J.* **24**(4), 17–31 (1991)
13. Anisi, D.A., Skourup, C.: A step-wise approach to oil and gas robotics. In: *Proceedings of the 1st IFAC Workshop on Automatic Control in Offshore Oil and Gas Production*, IFAC Proceedings Volumes, vol. 45, no. 8, pp. 47–52 (2012)
14. Yim, M., Roufas, K., Duff, D., Zhang, Y., Eldershaw, C., Homans, S.: Modular reconfigurable robots in space applications. *Auton. Robots* **14**(2), 225–237 (2003)
15. Maslowski, A.: Intelligent mobile robots supporting security and defense. In: *Proceedings of the International Conference IMEKO TC-17 Measurement and Control in Robotics*, NASA Space Center, Huston, Texas, USA (2004)
16. Carroll, D.M., Nguyen, C., Everett, H.R., Frederick, B.: Development and testing for physical security robots. In: *Proceedings of SPIE 5804, Unmanned Ground Vehicle Technology VII*, 550 (2005). doi:[10.1117/12.606235](https://doi.org/10.1117/12.606235)
17. Parker, L.E.: Path planning and motion coordination in multiple mobile robot teams. In: *Encyclopedia of Complexity and System Science*, pp. 5783–5800 (2009)
18. Galceran, E., Carreras, M.: A survey on coverage path planning for robotics. *Rob. Auton. Syst.* **61**(12), 1258–1276 (2013)
19. Hameed, I.A.: Intelligent coverage path planning for agricultural robots and autonomous machines on three-dimensional terrain. *J. Intell. Rob. Syst.* **74**(3–4), 965–983 (2014)
20. Hameed, I.A., Bochtis, D., Sørensen, C.A.: An optimized field coverage planning approach for navigation of agricultural robots in fields involving obstacle areas. *Int. J. Adv. Rob. Syst.* **10**, 231–2013 (2013). InTech
21. Lekkas, A.M., Dahl, A.R., Breivik, M., Fossen, T.I.: Continuous-curvature path generation using fermat's spiral. *Model. Ident. Control* **34**(4), 183–198 (2013). ISSN 1890–1328
22. Hameed, I.A., Sorrenson, C.G., Bochtis, D., Green, O.: Field robotics in sports: automatic generation of guidance lines for automatic grass cutting, striping and pitch marking of football playing fields. *Int. J. Adv. Rob. Syst.* **8**(1), 113–121 (2011). InTech, ISSN 1729–8806
23. FIFA: Laws of the game 2015/2016, pp. 1–140 (2016). <http://www.fifa.com>
24. Hameed, I.A., Bochtis, D.D., Sørensen, C.G., Nøremark, M.: Automated generation of guidance lines for operational field planning. *Biosyst. Eng.* **107**(4), 294–306 (2010)

25. Dubins, L.E.: On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangents. *Am. J. Math.* **79**, 497–516 (1957)
26. Shkel, A.M., Lumelsky, V.: Classification of the dubins set. *Rob. Auton. Syst.* **34**(4), 179–202 (2001)