ELECTRONICS AND ELECTRICAL ENGINEERING
ISSN 1392 – 1215 2008. No. 1(81)
ELEKTRONIKA IR ELEKTROTECHNIKA

T125 AUTOMATION, ROBOTICS
AUTOMATIZAVIMAS, ROBOTECHNIKA

Vector Marks versus Potential Field

S. Bartkevičius
Department of Theoretical Electrical Engineering, Kaunas University of Technology,
Studentų str. 48, LT-51367 Kaunas, Lithuania; phone: +370 37 300253; e-mail: Stanislovas.Bartkevicius@ktu.lt
V. Baranauskas, K. Šarkauskas
Department of Control Engineering, Kaunas University of Technology,
Studentų str. 48-107, LT-51367 Kaunas, Lithuania; phone: +370 61025585, e-mail Kastytis.Sarkauskas@ktu.lt

Introduction

Real time navigation of mobile robots requires knowing of necessary path or representation of the environment, robots move. Planned path usually concludes by a sequence of points in the environment. Representation of an environment is more complicated, but allows flexible navigation – a robot calculates its path dynamically. Popular representation of an environment is usage of means of electrostatic field [1] or potential functions [2, 3]. Usage of electrostatic field, when all obstacles have the same potential and only a goal point has another one, is very attractive, because all gradient lines leads to the goal and a robot moving along a gradient line always reaches the goal. On other hand, gradient line often is far from shortest possible path. Fig. 1 illustrates this, where obstacles are blank circles, goal – black circle and robot – square. Curves represent gradient lines of this configuration.

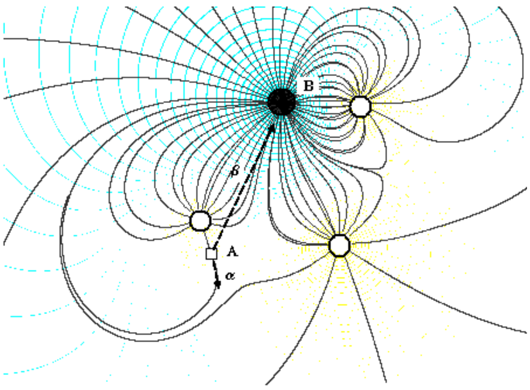

Fig. 1. A robot A moves along the gradient α although the shortest path is direct line β to the goal B

Another weakness of usage of electrostatic field is that only rather simple configurations of obstacles can behave analytical solution. Numerical methods, finite elements, for example, are applied then, but they requires large amount of nodes, field points, to use to ensure acceptable accuracy of a representation.

Usage of artificial potential functions decreases amount of calculations, but may cause deadlocks [2, 4] and special methods to search an exit from them are required. Representation of an environment by potential field also causes oscillating movement in narrow passages.

The weaknesses mention before stimulate to search for new methods of an environment representation. Requirements for these methods:

1. Ability to define a possible path to the goal or detect absence of it from any point of represented space;
2. The found path is near to optimal (shortest *etc.*);
3. Representation of an environment do not requires large amount of data.
4. An algorithm to calculate current step of robot is rather simple.

The purpose of this paper is to introduce a new representation of an environment, preserving positive features of usage of potential fields and requiring significantly less data necessary to calculate each step of a robot.

Shortest path

Let us treat a robot as a material point for the first approach. Optimal path means the shortest possible in this paper. The shape of the shortest path, for example, from an initial position A to the goal D (Fig. 2) depends on number of obstacles and their form. This path consists of sections of bitangential lines (FG , HJ) or segments of curves (GH), except of terminal fragments. The terminal fragments of the path are sections of tangents to appropriate obstacles and can change, if a position of a terminal point changes. It is evident, that change of position of an initial point (A, B, C) does not change significant part of the path. Therefore, a path is rather an attribute of the goal D. On other hand, there is a space where placed initial positions (A, B) of a robot require varying only one, initial, fragment of the shortest path.

A robot moving from initial position needs only a touch point to an obstacle, F or G, to detect direction of movement.

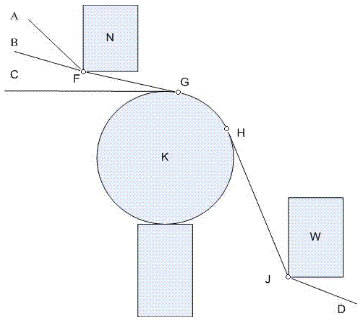

Fig. 2. The shortest path from initial positions (A, B, C) to the goal D . N, K, W – obstacles

Vector marks

The touch point of a shortest path and an obstacle represented by convex polygon is one of the vertexes of the obstacle. It is possible that a fragment of the shortest path coincides with an edge of an obstacle, Fig. 3, but even at this special case robot A needs only one convex D (robot E needs convex C) to determine direction.

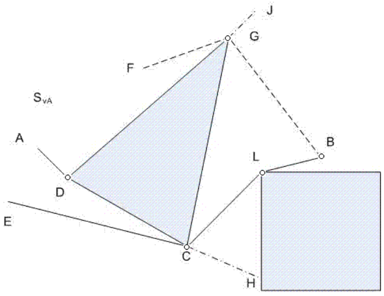

Fig. 3. The fragment of the shortest path coincides with edge of an obstacle

Robot needs a touch point (Fig.2. point G) to know to bypass obstacles represented by curves, circle for example. Anyway, only one visible point of a path is required to determine direction of the movement. Vertexes of polygons are fixed points, but touch points must by calculated dynamically.

It is not enough to know vertex or touch point. Vertexes D and G are visible from position A and F both, but they belong to different shortest ways. Therefore, visible vertexes must have weight depending of distance from the goal.

Let's designate the space occupied by obstacles as a set of points S_{obst} and space free of obstacles - S_{free}, a subspace, all points of which are visible from a vertex or touch point, belonging to a path, $- S_v$. The subspace S_{vD}, associated with vertex D is shown in Fig. 3. This subspace is limited by lines DH and DJ . An environment is fully defined by k subspaces, associated with k points, if the following condition is fulfilled:

$$S_{free} = \bigcup_{n=1}^{k} S_{vn} \ . \qquad (1)$$

If obstacles are convex polygons and all shortest paths are known this condition can de fulfilled easily, but if obstacle is not a polygon a touch point is not defined a priory, (Fig. 2, start position C , touch point G). Usage of visible lines can solve this problem. The authors propose to use a set of virtual vectors oriented along the nearest fragment of a path (bitangential line section) with head in vertex or touch point (Fig. 4). Let us name these vectors – vector marks.

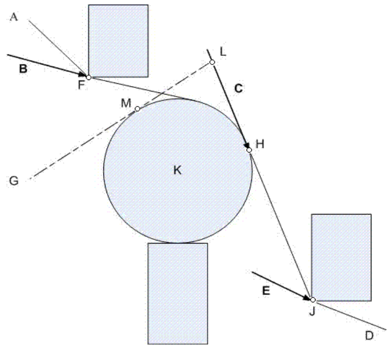

Fig. 4. Vector marks (***B, C*** and ***E***)

A set of vector marks is associated with a goal point and structured as an ordered tree, because there is possible that some optimal paths, depending on initial position of a robot, to the same goal exist. To define properties of vector marks let us define "visibility" – a point p_d is visible from point p_0 if a segment of line connecting these points exists and all points of this segment belongs to S_{free}:

$$\forall p \in l(p_0, p_d) \ \in S_{free}, \qquad (2)$$

where $l(p_0, p_d)$ denotes line connecting points p_0 and p_d . Properties of vector marks and visibility:

1. All vector marks lie in the space S_{free} .
2. Any vector mark is elongation of bitangential linear fragment of a path.
3. A mark is visible if there is at least one visible point of the mark.
4. From any point of S_{free} at least one of vector marks is visible. This property follows from condition (1).
5. Any mark has associated attributes; the main of them is *weight –total distance, along the* path *the mark associates, from the head of the mark to the goal point.*

Vector marks allow dynamic detecting of the nearest point of the shortest path (vertex or touch point). For example, robot at position G (Fig 4.) do not see touch point H , but it can see the point of vector mark L and start movement towards it, dynamically correcting direction of movement as new points of the mark, near the touch point H , become seen.

Algorithm of movement

To detect direction of movement robot needs to select only one vector mark. Vector marks are weighted, so, robot must select an appropriate vector mark according its weight. Weight of a mark, as mentioned before, is total distance from the goal point and head of the mark, but distances from initial position of the robot and visible point of mark and from this point to the head of the vector mark do not belong to the optimal path. To ensure the shortest path, robot must try to see a point of the vector mark as close to its head as possible. It means, that point robot see (Fig.4

point L), lies on tangent line connecting robot position G (Fig.4) with the vector mark and touching the obstacle K at point M. Therefore, criterion λ to select an appropriate vector mark from k visible may be:

$$\lambda = \min_{n=1}^{k}\left(W_n + \|\vec{p}_{nv} - \vec{p}_r\| + \|\vec{p}_{nh} - \vec{p}_{nv}\|\right),\ n \in \mathrm{N}, \qquad (3)$$

here W_n – weight of n-th vector mark, p_{nv} – visible point of the mark, p_r – position of robot, p_{nh} – head point of the vector mark, $\vec{p}_{nv}, \vec{p}_r, \vec{p}_{nh}$ – vectors from the beginning of coordinates to an appropriate point.

After an appropriate vector mark selected the robot R starts moving (Fig. 5) towards the mark along the line connecting initial position and visible point A.

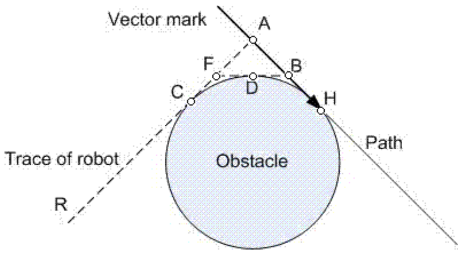

Fig 5. Correction of robot trace

The direction of movement needs of correction, when touch point C passed. The robot from point F looks for a new visible point B as near as possible to head of the vector mark H and moves towards point B. Direction of movement changes again when new touch point D passed. Repeating this process of correction robot can enter the predefined path reach or pas the head point of the vector mark. It is rather complicated to detect touch point to an obstacle. Therefore, it seems expedient to do the correction of direction on each step of robot. Such the simplified algorithm of movement represented in pseudo code is:

$\vec{\sigma} \leftarrow$ unit vector of direction of the vector mark,
$p_h \leftarrow$ head point of the mark,
$p_t \leftarrow$ tail point of the mark,
$a \leftarrow$ step along the vector mark,
$h \leftarrow$ step of robot,
$p \leftarrow$ start position (point) of robot;
while (p_h not reached or passed)
  $p_v \leftarrow p_h$ **;**
  while (p_v not visible)
    $\vec{p}_v \leftarrow \vec{p}_v - \vec{\sigma} \cdot a$
  end
  calculate unit vector $\vec{v}$ of direction of line from p to p_v,
  if $\|\vec{p}_v - \vec{p}\| \le h$,
    $\vec{p} \leftarrow \vec{p}_v$
  else
    $\vec{p} \leftarrow \vec{p} + \vec{v} \cdot h$
  endif
end

It is evident, that the trace of robot is near the shortest path, if step of robot movement h is not too large. If length of some different paths is similar and exactly the shortest path is required, the criterion (3) for selection of visible vector mark is rather coarse. The corrected one, based on robot movement algorithm, may be useful:

$$\lambda_C = \min_{n=1}^{k}\left(W_n + \sum_{m=1}^{z}\|\vec{p}_{m-1} - \vec{p}_m\|\right),\ n \in \mathrm{N},\ m \in \mathrm{N}, \qquad (4)$$

here z – number of steps required to reach or pass the head point of a vector mark, p_m - point, at which correction of supposed robot movement direction is done.

Evaluation of dimensions of a robot

Real robot is not a mathematical point. Therefore, a path of a robot must be planned evaluating dimensions of a robot. The simplest way to do it is to move some points of a path, planned as described before, so that ensure required distance between the robot and obstacles bypassed. For example, point F (Fig. 6) substitutes point D, so that distance from obstacle and new fragment of path AF is required gap d. This case represents situation, when obstacle and robot are on different sides of the vector mark, the head of mark is visible and the necessary gap g warranted replacing the touch point by new point F placed on the normal to vector mark D. If obstacle and robot are on the same side of a vector mark, a necessary gap g can be warranted replacing real visible point of mark C by another point G, for example.

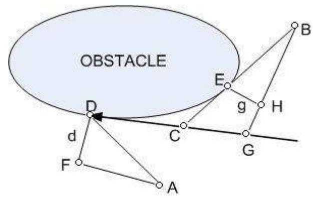

Fig. 6. Clearances between obstacle and robot. A, B – places of robots

Therefore, robot moves along the line AF instead of AD. It is necessary to introduce one another attribute of a vector - mark qualifying on what side, versus orientation of vector mark, an obstacle placed. Let us call this attribute *side.* The manner of evaluation of dimensions of robot lets do it dynamically. This is important, because overall dimensions of a robot can change depending of load carrying by a transportation robot, for example.

Example of simulated movement

The idea of vector marks tested by means of simulation using program package CENTAURUS CPN [5]. The algorithm of movement of robot realized as colored Petri net in attempt to make any correction or improvement easy and ostensive.

To define visibility of marks the function checking if the “beam” - straight line drawn from current robot position along selected direction, crosses an obstacle written. The return value of the function is nearest cross point, if an obstacle crossed, and arbitrary “far” point, if not. Such function is sufficient to detect visible vector marks and obstacles.

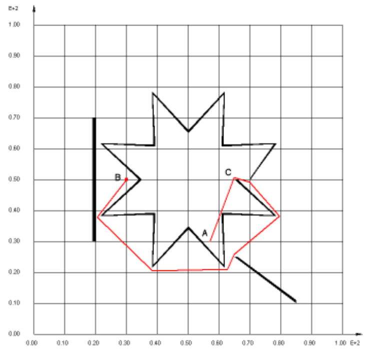

Fig. 7. Path of robot from initial place of robot A to goal B in complicated environment

The example (Fig. 7) shows the path of robot in complicated environment. There are sharp turns (point C) in this case. It is necessary to know the angle of turn beforehand to adopt speed of robot. This angle can be calculated if the point next to C should be known. The control system of a robot must be able to calculate at least one point of turn ahead. The algorithm described before allows calculation of the entire path and smoothing of it using splines, for example, or another smoothing method.

Conclusions

New representation of indoor environment as a set of vector marks is proposed. Vector marks must meet such requirements:

a) number of marks and setout of them must ensure at least one mark visible, from each point of environment;
b) mark must lay on tangent to obstacle, with the head in touch point and direction towards the goal point;
c) four attributes associated with each mark: *coordinates of head and tail points, weight value and side indicator*;
d) *weight value* is proportional to distance from the head point of mark and goal point;
e) *side* of mark shows on what side of vector obstacle lays.

Using of vector marks lets move a robot by the path approximate to the shortest way.

References

1. **Pimenta L.C.A. et al.** Robot Navigation based on Electrostatic Field Computation // IEEE Transactions on magnetics. – April 2006. – Vol. 42, No. 4. – P. 1459–1462.
2. **Cosio F.A.** Autonomous robot navigation using Adaptive Potential Fields // Mathematical and computer modeling. – 2004. – No.40. – P. 1141–1156.
3. **LaValle S. M.** Planning Algorithms. – Cambridge University Press, 2006.
4. **Ge S. S. and Cui Y. J.** New Potential Function for Mobile Robot Path Planning // IEEE Transactions on robotics and automation. – October 2000. – Vol. 16, No. 5. – P. 615–620.
5. **Baranauskas V., Bartkevičius S., Kragnys R., Šarkauskas K.** Modelling Control systems with Coloured Petri Nets Using Global Variables // Proceedings of intern. conf. Electrical and Control Technologies – 2006. – Kaunas. – P. 306 –309.

Submitted for publication 2007 09 10

S. Bartkevičius, V. Baranauskas, K. Šarkauskas. Vector Marks versus Potential Field // Electronics and Electrical Engineering – Kaunas: Technologija, 2008. – No. 1(81). – P. 49–52.

In navigation of robots in indoor environment potential field, electrostatic or based on artifical potential functions, are often used. There are known, except of advantages, some weeknesses of this usage – found paths are not optimal, large amount of data is required, possible deadlocks *etc*. The new method of representation of an environment, preserving the main advantage of usage of potential fields – possibility to define path of movement dynamicali and do not requirening complicated calculations during movement and ensuering paths near the shortest proposed. Robot moves orienting only by the set of virtual vectors (vector marks) associated with the goal pont. Vector marks touches obstacles and are an elongations of fragments of the shortest path. They are placed so, that at least one vector mark is visible from any position of a robot. Attributes of vector marks and the algorithm of robot orientation are presented. Ill. 7, bibl.5 (in English; summaries in English, Russian and Lithuanian).

С. Барткевичюс, В. Баранаускас, К. Шаркаускас. Векторные метки вместо потенциального поля // Электроника и электротехника. – Каунас: Технология, 2008. – № 1(81). – С. 49–52.

В навигации роботов в известной обстановке часто используются потенциальные поля, электростатические или основанные на искусственных потенциальных функциях. Кроме преимуществ использование потенциальных полей обладает и недостатками – найденные трассы далеки от оптимальных, требуется обработка большого количества данных, возможны тупики. Предлагается новый способ сохраняющий основное преимущество использования потенциальных полей – возможность динамического определения трассы движения и не требующий сложных вычислений во время движения, гарантирующий трассы близкие наикратчайшим. Робот ориентируется, пользуясь системой виртуальных векторов, векторных меток, связанных с точкой цели. Векторных метки касаются препятствий и являются продолжениями фрагментов наикратчайшего пути. Они располагаются так, что из каждого возможного местоположения робота видна, по крайней мере, одна векторная метка. Описаны атрибуты векторных меток и алгоритм ориентации робота. Ил. 7, библ.5 (на английском языке; рефераты на английском, русском и литовском яз.).

S. Bartkevičius, V. Baranauskas, K. Šarkauskas. Vektorinės žymės vietoj potencialinio lauko // Elektronika ir elektrotechnika. – Kaunas: Technologija, 2008. – Nr. 1(81). – P. 49–52.

Robotų navigacijoje žinomoje aplinkoje dažnai naudojami potencialiniai laukai – elektrostatiniai ar grįsti dirbtinėmis potencialinėmis funkcijomis. Be privalumų, potencialinių laukų taikymas turi ir trūkumų – randamos trasos nėra optimalios, galimos aklavietės, reikalingi dideli apdorojamos informacijos kiekiai ir t. t. Darbe siūlomas naujas aplinkos, kurioje juda robotas, aprašymo būdas, išlaikantis pagrindinį potencialinių laukų taikymo privalumą – galimybę dinamiškai rasti judėjimo trasą ir nereikalaujantis didelių skaičiavimų roboto judėjimo metu, užtikrinantis, kad rastos robotų trasos būtų artimos optimalioms – trumpiausioms. Robotas orientuojasi naudodamasis virtualių vektorių (vektorinių žymių) sistema, susijusia su tikslo tašku. Vektorinės žymės liečia kliūtis ir yra trumpiausio kelio fragmentų tęsiniai. Jos išdėstomos taip, kad iš bet kurios roboto pozicijos būtų matoma bent viena žymė. Darbe aprašyti vektorinių žymių atributai ir roboto orientavimosi algoritmas. Il. 7, bibl. 5 (anglų kalba, santraukos anglų, rusų ir lietuvių k.).

DOI: 10.5755/j02.eie.11033