Proceedings of the
2007 IEEE Intelligent Vehicles Symposium
Istanbul, Turkey, June 13-15, 2007

Elliptical trajectories for non-holonomic vehicles

F. Bampi, A. Scalzo, R. Zaccaria and C. Zordan

Abstract— **A manipulation device can be installed on a mobile robot. In this case, the robot should be able to approach the objects to be manipulated from a given direction, suitable for this purpose. If the kinematic structure of the robot is not holonomic, this problem can be very hard to solve. In this paper, it is described a navigation algorithm capable of driving a car-like robot (i.e. characterized by slow steering motion with a minimum radius of curvature) to a given point with a given orientation, whatever its initial position and orientation.**

I. Introduction

Consider a manipulation device or a tool of some kind mounted on a mobile robot, which purpose is to interact with the objects in the environment. In the simplest case it can be an automated fork-lift vehicle, whilst in the most complex ones it can be a mobile robot carrying robotic arms provided with hands or tools. In order to interact with the objects by its tools, it is very likely that the approaching direction is fundamental in the execution of its task (for example, we can still think about a fork-lift vehicle). The kinematic structures of robotic vehicles of this kind are rarely characterized by holonomic motion. Instead, specially for large and heavy vehicles, it is more likely that they have a car-like structure, i.e. characterized by slow steering and minimum radius of curvature. Many algorithms [1] have been proposed to solve this problem. The most famous is probably the “2 or 3 balls” manoeuvre algorithm, but it can’t generate trajectories having continuous curvature. Continuous curvature in trajectories is necessary if it is required to follow them with a slow steering car-like robot without stopping it during the motion. In [2] Scheuer and Frauchard solve this problem using a set of 8 templates. The algorithm presented in this paper achieve this goal using an ellipse or a combination of a circle and of an ellipse to generate the trajectory of the robot from a starting position and orientation to a goal. In the following sections, the theoretical fundamentals of this algorithm will be discussed, and then in the last section it will be shown the integration of the algorithm in the middle-level control loop of the robot, together with some results of the application of this technique. This implies that the proposed algorithm belongs to path planning and kinematic control techniques, and not to low level dynamic control, which must be implemented an inner dynamic control loop.

Manuscript received January 14, 2007.

Franco Bampi is with Dipartimento di Ingegneria della Produzione, Termoenergetica e Modelli Matematici, University of Genova, 16100 Genova, Italy, e-mail: bampi@diptem.unige.it

Alessandro Scalzo is with Dipartimento di Informatica, Sistemi e Telematica, University of Genova, 16100 Genova, Italy, e-mail: ale.scalzo@atin.it

Renato Zaccaria is with Dipartimento di Informatica, Sistemi e Telematica, University of Genova, 16100 Genova, Italy, e-mail: renato@dist.unige.it

Clara Zordan is with Dipartimento di Ingegneria della Produzione, Termoenergetica e Modelli Matematici, University of Genova, 16100 Genova, Italy, e-mail: zordan@dibe.unige.it

II. Conical trajectories

A. Formalization of the problem

Assume that the robot starts from the point P and arrives at the point O. Assume also that the unit vectors $\mathbf{t}_i$ and $\mathbf{t}_f$, denoting the initial and final robot orientation, respectively, are known. Choose a frame system (O, x, y) in such a way that the x axis is oriented as $\mathbf{t}_f$ and, letting $P = (X, Y)$, the y axis is oriented so that $Y > 0$, as in Fig. 1. Finally, denote by τ the tangent of the angle between the *x* axis and $\mathbf{t}_i$.

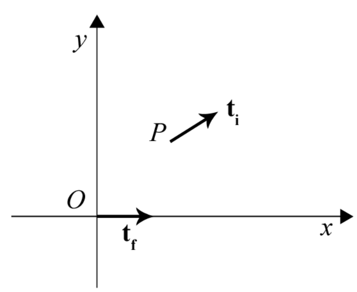

Fig. 1. problem reference frame

The idea is that the trajectory of the robot is calculated by joining P and O with a conic curve having $\mathbf{t}_i$ and $\mathbf{t}_f$ as tangent unit vectors in P and O, respectively. From a mathematical point of view, the problem consists in finding a conic of the form

$$f(x, y) = x^2 + by^2 + 2cxy + dx + ny + p = 0$$

constrained by the conditions

$$f(X, Y) = 0$$
$$f(0,0) = 0$$
$$f_x(X, Y) = -\tau f_y(X, Y)$$
$$f_x(0,0) = 0$$

B. Conics classification

As an immediate consequence, we find $p = 0$ and $d = 0$, while calculation shows that

$$b = \frac{X^2}{Y^2} - \frac{n}{L} \qquad c = -\frac{X}{Y} + \frac{\tau n}{2L}$$

where the quantity $L = Y - \tau X$ must be non null, otherwise the conic is degenerate.

A straightforward calculation allows to classify the conic in terms of the parameter n. Precisely, when $\tau \neq 0$, it can be set

$$n_p = -\frac{4L^2}{\tau^2 Y} < 0$$

that yelds the following classification

- $n < n_p$: hyperbola
- $n = n_p$: parabola
- $n_p < n < 0$: ellipse
- $n = 0$: degenerate conic
- $n > 0$: hyperbola

The case $\tau = 0$ yields the classification

- $n < 0$: ellipse
- $n = 0$: degenerate conic

$n > 0$: hyperbola

III. Choosing the Conic

The results found so far are very general. The task is now that of choosing a conic that allows the robot to start in P and arrive at O following a continuous trajectory. It is evident that hyperbola, having two separated branches, is unsuitable for such purpose. On the other hand, parabola is a open curve: therefore it does not guarantees that the orientations of $\mathbf{t}_i$ and $\mathbf{t}_f$ are correct. In other words this means that the robot must invert its position before starting and, eventually on arriving, fact that is not always possible.

To avoid as far as possible such a kind of drawbacks the possible trajectories are restricted to elliptic curves. To sum up previous results to such a case, the family of elliptic solutions to the problem are parametrized with a constant γ by setting

$$n = -\gamma \frac{L^2}{Y} \qquad 0 < \gamma < \frac{4}{\tau^2}$$

which includes also the case $\tau = 0$. Accordingly, all the results for elliptical solutions can be summed up as follows

$$f(x, y) = x^2 + by^2 + 2cxy + ny = 0$$

where

$$L = Y - \tau X \neq 0 \qquad (Y > 0)$$

$$n = -\gamma \frac{L^2}{Y} \qquad 0 < \gamma < \frac{4}{\tau^2}$$

$$b = \frac{X^2}{Y^2} + \frac{\gamma L}{Y} \qquad c = -\frac{X}{Y} - \frac{\gamma \tau L}{2Y}$$

In Fig. 2 is shown a family of elliptic paths joining two robot start and goal positions.

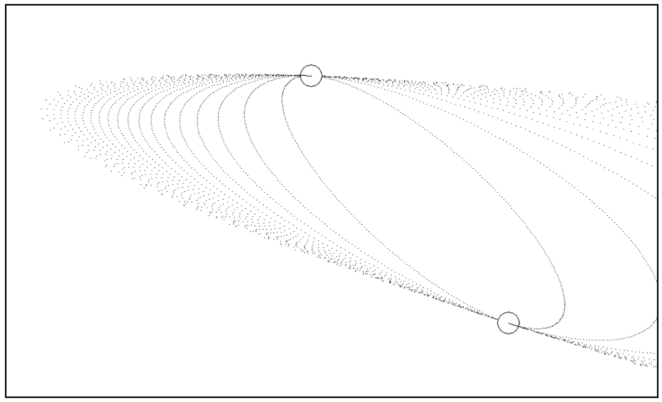

Fig. 2. Elliptical trajectories from start to goal

IV. The Problem of the Curvature

Denoting by α and β, $\alpha < \beta$, the semi axes of an ellipse, it can be proved that

$$\alpha = \sqrt{\frac{\gamma L^2}{(4 - \gamma \tau^2)} \frac{1}{A}} \qquad \beta = \sqrt{\frac{\gamma L^2}{(4 - \gamma \tau^2)} \frac{1}{B}}$$

where the positive quantities $A > B$ are given by

$$A = \frac{b+1}{2} + \frac{\sqrt{(b-1)^2 + 4c^2}}{2}$$

$$B = \frac{b+1}{2} - \frac{\sqrt{(b-1)^2 + 4c^2}}{2}$$

The curvature radius ρ varies in the range

$$\rho_{\min} \leq \rho \leq \rho_{\max}$$

where

$$\rho_{\min} = \frac{\alpha^2}{\beta} = \sqrt{\frac{\gamma L^2}{4-\gamma\tau^2}\frac{B}{A^2}}$$

$$\rho_{\max} = \frac{\beta^2}{\alpha} = \sqrt{\frac{\gamma L^2}{4-\gamma\tau^2}\frac{A}{B^2}}$$

Since, in general, a robot can follow trajectories whose curvature radius is greater than a given value ρ_0 determined by its mechanical structure, a condition should be fulfilled that there exists almost a value of γ such that $\rho_{\min} \le \rho_0$. In this case a robot trajectory is found. Otherwise, the procedure described in next section must be applied.

V. Adjusting Elliptical Trajectories

A. Initial and final orientation

Besides the case outlined at the end of previous section, an elliptical trajectory could not be found also because the orientations of $\mathbf{t}_i$ and $\mathbf{t}_f$ are wrong in the sense that starting from P with direction $\mathbf{t}_i$ the robot arrives in O with the orientation opposite to $\mathbf{t}_f$. To know in advance whether such is the case, we proceed in the following way. Draw two straight lines from P and O with direction $\mathbf{t}_i$ and $\mathbf{t}_f$, respectively, and denote by Z their intersection point. Define two scalars r and s by the formulas

$$Z - P = r\mathbf{t}_i$$
$$Z - O = s\mathbf{t}_f$$

So, correct orientation corresponds to condition $rs < 0$ (as in Fig. 3) whereas the case $rs > 0$ requires to adjust the orientation before start.

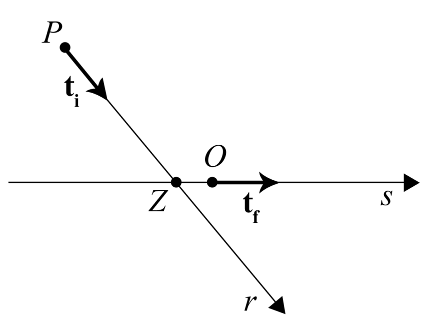

Fig. 3. Existence of an elliptical trajectory

Adjustment can be achieved by making the robot follow a counterclockwise circle of radius R (to be chosen later) until it reaches the highest point Q of the circle with orientation $-\mathbf{t}_f$. Therefore, the next arc of trajectory is an ellipse whose initial curvature radius is R. In this way the parameter γ is calculated as a function of R and R is chosen so that $\rho_{\min} \le \rho_0$.

B. Calculation of Q

Introduce the vector $\mathbf{n}_i$ by rotating counterclockwise $\mathbf{t}_i$ of the angle $\pi/2$ and set $\mathbf{n}_i = (n_1, n_2)$. It is immediate to prove that

$$Q = (X_Q, Y_Q) = (X + Rn_1, Y + R(n_2 + 1))$$

C. Imposing the curvature radius

After the preliminary circular trajectory, the robot is in a position that allows to reach the goal with a elliptic trajectory. The new starting condition is the position Q with orientation $-\mathbf{t}_f$, that means $\tau = 0$. Previous formulas become

$$b = \frac{X_Q{}^2}{Y_Q{}^2} + \gamma \qquad c = -\frac{X_Q}{Y_Q}$$

where now $0 < \gamma < +\infty$, and

$$\alpha = \frac{Y_Q}{2}\sqrt{\frac{\gamma}{A}} \qquad \beta = \frac{Y_Q}{2}\sqrt{\frac{\gamma}{B}}$$

Obviously, A and B are calculated in terms of the coordinates of Q with $\tau = 0$.

Introduce the new coordinates

$$x' = [\lambda(x - X_Q/2) + \mu(y - Y_Q/2)]/\sqrt{\lambda^2 + \mu^2}$$
$$y' = [\lambda(y - Y_Q/2) - \mu(x - X_Q/2)]/\sqrt{\lambda^2 + \mu^2}$$

where λ and μ are connected by relation

$$\mu = \frac{b - 1 + \sqrt{(b-1)^2 + 4c^2}}{2c}\lambda$$

The curvature radius in a point of the ellipse with coordinate (x, y) is given by

$$\rho = \frac{(\alpha^4 y'^2 + \beta^4 x'^2)^{3/2}}{\alpha^4\beta^4}$$

so that whatsoever condition on it ca be imposed.

VI. A Particular Application

In order to show an explicit calculation, consider the case where the initial position of the robot has $X = 0$ and

$\mathbf{t}_i = \mathbf{t}_f$ which implies $\tau = 0$. Following the procedure of section 5 it results

$$Q = (0, Y + 2R) \quad \mathrm{b} = \gamma \quad \mathrm{c} = 0$$

Note that $c = 0$ is not a problem in that, due the symmetry of the situation, we need not use the new coordinate x' and y'. Indeed we start from Q with an ellipse which has one of the two axes of length $Y + 2R$ and coincident with the y-axis; therefore the curvature radius in Q is either minimal or maximal. However, the condition that the curvature radius in Q is exactly R implies that the curvature radius in Q must be necessarily minimal. It means that the vertical axis must be the major axis of the ellipse. Consequently, α and β must be chosen as

$$\alpha = \frac{Y + 2R}{2}\sqrt{\gamma} \qquad \beta = \frac{Y + 2R}{2}$$

with $\gamma < 1$. Condition $\rho_{\min} = R$ provides the value of γ, namely

$$\gamma = \frac{2R}{Y + 2R}$$

It is guaranteed that $\rho_{\min} \le \rho_0$ by letting $R \ge \rho_0$.

VII. The Control Loop

If $\mathbf{t}$ is the actual orientation of the robot in the actual position $\mathbf{p}$, the kinematics of the robot can be described by

$$\dot{\mathbf{p}} = \mathit{speed}\ \mathbf{t}$$
$$\dot{\mathbf{t}} = \mathit{speed}\ k\ \mathbf{n}$$

where $k = 1/\rho$ i.e. the curvature of the elliptical or circular trajectory in $\mathbf{p}$ and $\mathbf{n} = \mathbf{R}(\pi/2)\mathbf{t}$. It is obvious that the angular velocity of the robot is $jog = speed\ k$. The *speed* parameter is free, and given by the outside.

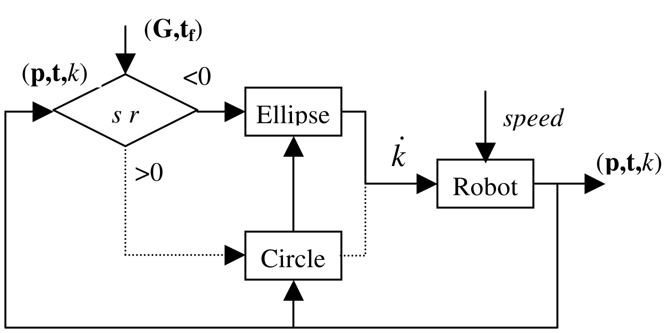

Fig. 4. Robot control loop

Notice that the entire process described in the previous sections is iterated at every control cycle. In this way, the algorithm can correct different kinds of disturbs that affect the robot motion. From this point of view, also if the inner dynamic control loop is not able to perfectly implement the trajectory generated by the outer trajectory generation loop, this can be assumed as a disturb that can be compensated at outer level.

As shown in the control schema (Fig. 4), the control variable in input to the robot is $\dot{k}$, the curvature time derivative (it is obvious that the robot steering angular speed can be expressed as a function of this quantity).

$\dot{k}$ can be calculated as follows: express the equation of the ellipse as

$$(\mathbf{x} - \mathbf{q})^t \mathbf{A}^{-1} (\mathbf{x} - \mathbf{q}) = 1$$

where q is the center of the ellipse and A is a definite positive matrix. Thus, define the transformation

$$\mathbf{x} = \mathbf{q} + \mathbf{T}\begin{bmatrix} R_1 & 0 \\ 0 & R_2 \end{bmatrix}\begin{pmatrix} \cos\vartheta \\ \sin\vartheta \end{pmatrix}$$

where T is the rotation matrix made of the eigenvectors of A, and R1 and R2 are the squared roots of the eigenvalues of A. In this way, the points of the ellipse represented by the previous equations are expressed as a function of the parameter ϑ.

Being

$$\mathbf{t} = \frac{\frac{\partial \mathbf{x}}{\partial \vartheta}}{\left|\frac{\partial \mathbf{x}}{\partial \vartheta}\right|} \qquad k\mathbf{n} = \frac{\frac{\partial \mathbf{t}}{\partial \vartheta}}{\left|\frac{\partial \mathbf{x}}{\partial \vartheta}\right|}$$

then

$$k = \frac{R_1 R_2}{\left(R_1^2 \sin^2\vartheta + R_2^2 \cos^2\vartheta\right)^{\frac{3}{2}}}$$

$$\dot{k} = -3\mathit{speed}\frac{R_1 R_2 \sin\vartheta\cos\vartheta}{\left(R_1^2 \sin^2\vartheta + R_2^2 \cos^2\vartheta\right)^3}$$

VIII. Conclusion

The following figure (Fig. 5) shows the trajectory generated by the method presented in this paper. The curve is continuous on the second derivative, thus guaranteeing that it is feasible by a slow steering vehicle at constant speed and without stops. Moreover, its radius of curvature is always greater than a given one, imposed by the geometry of the vehicle and by its constraints.

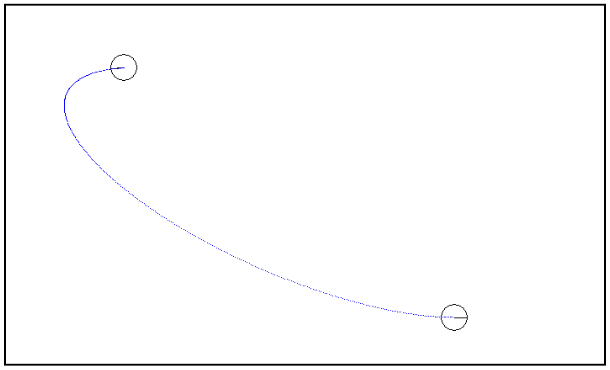

Fig. 5. Elliptical trajectory

In the second image (Fig.6), the circular trajectory (from point 1 to point 2) is followed by the robot until it reaches the condition to follow an elliptical trajectory (from point 2 to point 3). The left and right tangents of the curve in point 2 are equals.

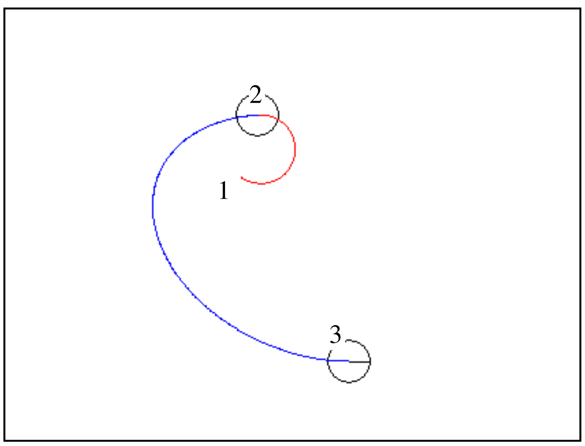

Fig. 6. Circular trajectory followed by elliptical trajectory

The algorithm presented in this paper is actually implemented in the robot Staffetta, an industrial and research middle-sized AGV (its weight is about 90 kg) employed in transportation and surveillance tasks in hospital and airport environments (Fig. 7). Its kinematics is redundant, with active steering and two independently motorized traction wheels. In the first 50 trials during the algorithm test, the robot reached the goal with an average error in final orientation of 3.2 degrees average, and a maximum error of 5.1 degrees.

Fig. 7. Robot Staffetta in hospital environment

REFERENCES

[1] Jean Claude Latombe, 1991, "Robot Motion Planning", Kluwer Academic Publisher.

[2] A. Scheuer, Th. Fraichard, 1997, "Continuous-curvature path planning for car-like vehicles", Proceedings of IROS 97.