Journal of Applied Sciences

ISSN 1812-5654

Journal of Applied Sciences 7 (13): 1854-1863, 2007
ISSN 1812-5654

Survey and Implementation on DSP of Algorithms of Robot Paths Generation and of Numeric Control for Mobile Robot

[1]Kais Bouallegue and [2]Abdessattar Chaari
[1]Department of Informatics, Higher Institute of Applied Sciences and Technology, City Taffala Ibn Khaldoun-BP n° 40-4003, Sousse, Tunisia
[2]National Engineering School of Sfax, Street of Soukra, BP: W, 3038, Sfax, Tunisia

Abstract: In this study, one propose to study a numeric type strategy permitting the generation of any shape of path in view of the scheduling of the trajectories for a car-like mobile robot where the planned motions considered are continuous sequences in the space of the robot. These paths are programmed in order to have some types of closed or open trajectories. One is interested in the motion control of the robot from an initial position to a final position while optimizing the consumed energy in its alternated circular motion on both sides of the segment joining these two points. In this study, one presents a new method based on a numeric approach conceived from the kinematics equations of the robot. This new technique of numeric, adaptive and dynamic control of the robot is implemented on DSP21065L of the SHARC family. This algorithm assures the robot control of an initial position of departure to a final position of arrival without the existence of obstacles.

Key words: Dynamical system, numeric control, car-like mobile robot, curvature, DSP SHARC

INTRODUCTION

During these last decades a sensitive increase of the interest is carried as well as to the mobile robotics in the domain of research that the one of the industry, in the goal of the implementation of the automated means. In this type of applications, the aimed objectives are very different from those of first time of the mobile robotics where the robot was very manageable and could not move that in an environment adapted to its perfectly known and unchangeable need. By consequence the mobile robots must have, actually, the perception capacities of the environment and the high-quality capacities for the decision making. Among those capacities, one can mention the scheduling of motion trajectories: The capacity to determine the path that allows the robot to move from a position to another while optimizing, on the one hand, the expense of energy during the motion and while avoiding, on the other hand, the obstacles of the environment (Scheuer and Laugier, 1998). In this contribution, one proposes to study a numeric type strategy allowing the generation of any shape of path in the purpose of the scheduling of the trajectories for a mobile robot where the motion planning is considered as a continuous sequence in the robot space. These paths are programmed in order to have some types of closed or open trajectories. One is interested in the control of the robot motion (Kumar *et al*., 1997) from an initial position to a final one with the optimisation of the consumed energy in its circular motion on both sides of the segment which joins these two points. In the literature, one can mention various techniques of approaches of trajectory paths scheduling such as the analytic and geometric techniques. Loumand demonstrated of analytic manner that a robot is governable in a small time provided that it makes some maneuvers (Laumond, 1987). Reeds and Shepps generalized the result obtained by Dubbins (Reeds and Shepp, 1990) in the case of the paths with maneuvers. Bonnard showed that a robot is governable in a small time if the algebra of Dregs which is generated by its control vectors is of maximal dimension (Lobry, 1981). Scheuer and Laugier had incorporated a new constraint into the problem of paths planning: The curvature derivative must be bounded to make the path smoother. In the opposite to Dubins' path, the curvature profile along the path has a trapezoidal shape and is continuous. Kanayama and Miyake proposed a pair of clothoid curves, whose curvature varies linearly with the arc length which allows t connect two straight lines segments with zero curvature (Liang *et al*., 2005).

In this study, we propose a new method based on a numeric approach conceived from the kinematics equations of the robot. This new technique of numeric, adaptive and dynamic control of the robots is

Corresponding Author: Kais Bouallegue, Department of Informatics, Higher Institute of Applied Sciences and Technology, City Taffala Ibn Khaldoun-BP n° 40-4003, Sousse, Tunisia Tel: 74 274 088 Fax: 74 275 595

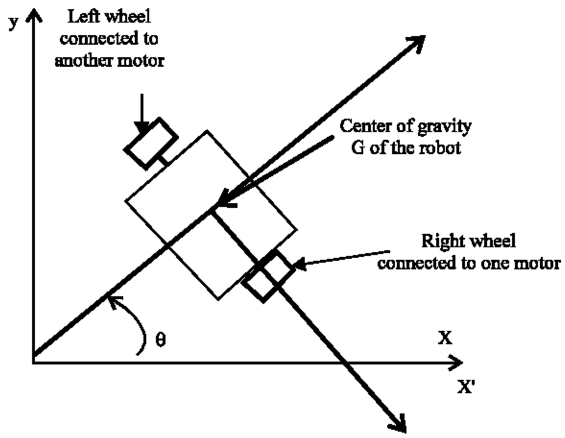


Fig. 1: Robot motion

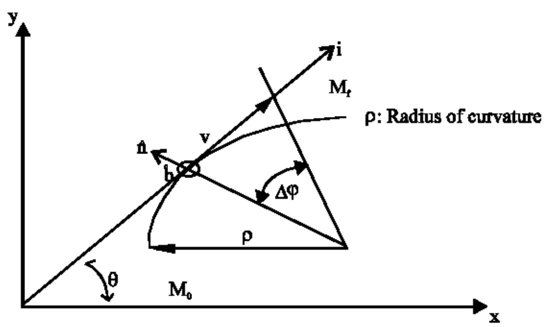


Fig. 2: Robot in rotation

implemented on DSP 21065L of the SHARC family. This algorithm assures the robot control from an initial position of departure to the arrival position without the existence of obstacles.

SCHEDULING OF TRAJECTORIES FOR A CAR-LIKE MOBILE ROBOT

Robot description: One considers a mobile robot noted B moving on a plane surface and is supported by two wheels and making point contact with the ground as shows the diagram of the Fig. 1. The car-like robot is modelled as a rigid body moving along a plane surface, and which has only one center of rotation under the perfect rolling condition and a wheel that must move along the normal direction to its axle. For a small time duration ΔT, the gravity center of the robot moves from M_0 to M_f by following the curved path (Fig. 2), with the tangential velocity v and the instantaneous center of rotation O (Hui *et al*., 2006). The configuration of the robot moving on a plane surface at every instant is defined by a triplet (x, y, θ).

Kinematics properties of the robot: While taking into account the geometric properties of the B robot, described in the previous section, we are going to consider its kinematics properties. The velocity of reference G (center of gravity) must remain perpendicular to the axis of equilibrium point and is directed therefore by the main axis of the robot. This constraint, named constraint of orientation, can be written as follow:

$$\dot{x}\sin\theta - \dot{y}\cos\theta = 0 \qquad (1)$$

where x and y are the coordinates of the robot position and $\dot{x}$ and $\dot{y}$ designate, respectively, the derivatives of x and y in relation to the time. In fact, the Eq. 1 can be simplified if one considers, separately, these derivatives. Indeed, if one notes v the velocity of the robot, one will have:

$$\begin{cases} \dfrac{dx}{dt} = v\cos\theta \\ \dfrac{dy}{dt} = v\sin\theta \end{cases} \qquad (2)$$

The rotation motion planning of the robot is done along a circular path. The radius of the curvature admits a lower boundary-mark imposed by the geometry of the robot. Therefore its tangential velocity must verify the following in equation:

$$v \geq \rho_{min}\dot{\varphi} \qquad (3)$$

where ρ_{min} designates the minimum radius of curvature. From the system of Eq. 2 and of the in equation (3), one can deduct the following relation:

$$(\dot{x})^2 + (\dot{y})^2 - (\rho_{min}\dot{\varphi})^2 \geq 0 \qquad (4)$$

The robot is conceived in such a way that each of wheels is driven individually by a connected motor through one gearbox and it rotates along the normal direction to its axle. Two DC motors are used to control the two wheels (Hui *et al*., 2006). It follows that the tangential velocity of the body of the car-like mobile robot is restricted and this is due to the condition of the motor constraint joined to the angular velocity. Let us assume that P is the power of the motor, N_{max} is the maximum rotational speed (in rpm), R is the radius of the wheels and C_R indicates the gearbox ratio. Thus, tangential acceleration γ must satisfy the following approximate relation:

$$\gamma \geq \frac{60P}{2\pi R \times C_R \times M \times N_{max}} \qquad (5)$$

Different approaches of the robots scheduling: The cases of paths scheduling for autonomous car-like mobile robots where the robots are capable to change directions, are restricted (Faichard and Scheuer, 2004). This problem has been extensively studied these last years and that met a variety of environmental constraints. The essential goal of this work is to generate a set of paths while using the mathematical equations established from the robot kinematics and geometric properties. In the objective of producing different paths forms or trajectories while using a DSP kit: ADSP 21065L EZS-KIT Lite, one must elaborate the dynamical numeric algorithms under a generalized shape.

Scheduling of the elliptic trajectories: Here, one is interested to the robot equations discretisation, in the order to elaborate the algorithms to implement on DSP and this for the generation of canonic trajectories. Let us leave from the equations system (2), one shows that the generation problem can be studied by the following general shape:

$$\begin{cases} \dot{x} = a\cos(\omega t) \\ \dot{y} = b\sin(\omega t) \end{cases} \tag{6}$$

with $\omega = 2\dfrac{\pi}{T}$,

where T is a signal period, t is the time, a and b are constants such as a∈ℝ* and b∈ℝ*. The discretisation of the equations system (6) gives:

$$\begin{aligned} x_{k+1} &= a\cos(\omega T_e k) \\ y_{k+1} &= b\sin(\omega T_e k) \end{aligned} \tag{7}$$

where T_e represents the sampling period and k designates the iteration number.
That allows to write:

$$\begin{cases} x_{k+1} = \alpha x_k - \dfrac{a}{b}\beta y_k \\ y_{k+1} = \dfrac{b}{a}\beta x_k + \alpha y_k \end{cases} \tag{8}$$

where α = cost (ωT_e), β = sin (ωT_e), x_0 and y_0 are the initial position coordinates. The system of recursive equations (8) permits to generate shapes of circular or elliptic trajectories, respectively, for a = b and a ≠ b.

Simulation examples: One took as initial values: x_0 = 0.5, y_0 = 0.5. The continuous signal period and the sampling period T and T_e are chosen, respectively, equal to 1s/5and to 1s/720. The Fig. 3-5 illustrate the forms of trajectories obtained for different values of a and b.

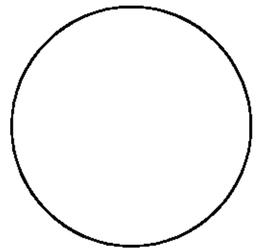


Fig. 3: Trajectory (a = 1, b = 1)

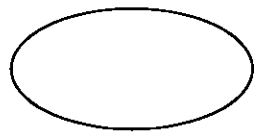


Fig. 4: Trajectory (a = 2, b = 1)

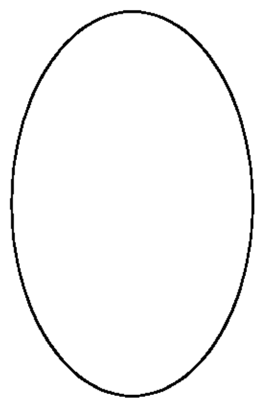


Fig. 5: Trajectory (a = 1, b = 2)

In this case, one describes the trajectory in a direct path while leaving from a point defined by the initial positions x_k and y_k to arrive to the point expressed by the final positions x_{k+1} and y_{k+1}. When one wants to describe the trajectory in the inverse sense, it is sufficient to determine and implement the equations system giving the recursive relationships under the following shape:

$$\begin{cases} x_k = \alpha x_{k+1} + \dfrac{a}{b}\beta y_{k+1} \\ y_k = -\dfrac{b}{a}\beta x_{k+1} + \alpha y_{k+1} \end{cases} \tag{9}$$

the simulation result is the same except that the trajectory path sense is, now, reversed. For adapting the equations system (8), one inserts a parameter, m, whose absolute value is chosen equal to the unit:

$$\begin{cases} x_{k+1} = \alpha x_k - m\frac{a}{b}\beta y_k \\ y_{k+1} = m\frac{b}{a}\beta x_k + \alpha y_k \end{cases} \quad (10)$$

The change of the m value (from +1 to -1 and vice versa) provokes necessarily the change of the trajectory path sense.

Generalization of the trajectories scheduling for mobile robots: The approach followed in order to lead this generalization of the scheduling consists in parameterizing the algorithm, that is defined by the Eq. 8. In fact, in the case of the robot (a = b = v), the system of Eq. 8 can be written:

$$\begin{cases} x_{k+1} = \alpha x_k - \beta y_k \\ y_{k+1} = +\beta x_k + \alpha y_k \end{cases} \quad (11)$$

with $\alpha = \cos(2\pi k\frac{T_e}{T})$ and $\beta = \sin(2\pi k\frac{T_e}{T})$,

if one assumes that $T_e/T = 1/N$, one can put α and β as follow:

$$\alpha = \cos(2\pi\frac{k}{N}) \quad \text{and} \quad \beta = \sin(2\pi\frac{k}{N})$$

In the goal to generate closed trajectories of complex shape; one adopts a structure of algorithm, analogous to the one definite by (11). Indeed, the choice of parameters values and allows to generate any forms of curvilinear trajectories.

Parameterisation of the algorithm: In the algorithm (11), which allows the trajectories scheduling, one chooses the value of α such as $\alpha = 1$ and parameterizing β to $\pm u_k$ of the following manner:

$$\begin{cases} x_{k+2} = x_{k+1} \pm u_k y_{k+1} \\ y_{k+2} = \pm u_k x_{k+1} + y_{k+1} \end{cases} \quad (12)$$

In the goal to have a great flexibility and easiness of the algorithm implementation; one had the idea to generate u_k by the same form of algorithm (11) in which α is maintained and β is replaced by β_1: Identical expression to the one of β with the only modification of k in k_1 in the following way:

$$\begin{cases} u_k = \alpha u_{k-1} + \beta_1 v_{k-1} \\ v_k = -\beta_1 u_{k-1} + \alpha v_{k-1} \end{cases} \quad (13)$$

with $\alpha = \cos(2\pi\frac{k}{N})$ and $\beta_1 = \sin(2\pi\frac{k_1}{N})$

Once the initial conditions (x_0, y_0, u_0 and v_0) are fixed, the choice of the trajectory form is only effected by the values given to k and k_1 for the four cases of possible configurations obtained by system (13). In addition, the idea of implementing the same structure of algorithm (11), presents the following advantages:

- Optimisation of space memory,
- Considerable gain in treatment and computing,
- High flexibility and easiness in the algorithm use (action on k and k_1).

The algorithm (12) could be written in the one of four possible configurations according to the choice of the parameters matrix

$$P = \begin{bmatrix} 1 & \pm u_k \\ \pm u_k & 1 \end{bmatrix}.$$

In each of these configurations, one adopted for the generation of u_k different values of k and k_1 in algorithm (13). The parameters matrix of (13) is as follow:

$$M = \begin{bmatrix} \cos(2\pi\frac{k}{N}) & \sin(2\pi\frac{k_1}{N}) \\ -\sin(2\pi\frac{k_1}{N}) & \cos(2\pi\frac{k}{N}) \end{bmatrix}$$

Results of the first algorithm configuration: One studied the implementation of the first algorithm structure that consists in fixing the matrix of parameters p to

$$P = \begin{bmatrix} 1 & u_k \\ -u_k & 1 \end{bmatrix}$$

in (12) and the parameters (k et k_1) of the matrix M in (13) according to the Table 1. The initial conditions x_0, y_0, u_0 and v_0 are chosen equal, respectively, to: 1, 0, et 1. In the implementation and for all trajectories curves, N is fixed to N = 1310.

The Fig. 6-11 illustrate the obtained results.

Results of the second algorithm configuration: One studied the implementation of the second algorithm structure (12), in which the matrix of parameters

$$P = \begin{bmatrix} 1 & -u_k \\ +u_k & 1 \end{bmatrix}.$$

The initial conditions are chosen equal to those adopted in the previous configuration while keeping the values of k and k_1 in matrix M identical to those of the Table 1. The Fig. 12-17 illustrate the obtained results.

Results of the third algorithm configuration: In the implementation of the third algorithm structure (12), the

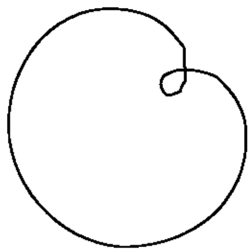


Fig. 6: Trajectory (k = 2, k1 = 2)

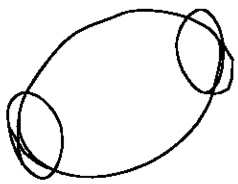


Fig. 7: Trajectory (k = 1, k1 = 2)

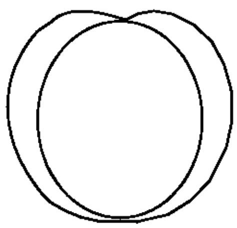


Fig. 8: Trajectory (k = 2, k1 = 1)

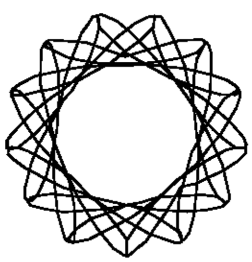


Fig. 9: Trajectory (k = 2, k1 = 13)

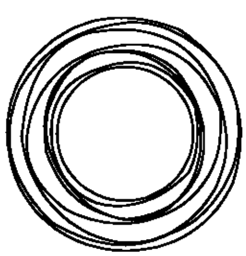


Fig. 10: Trajectory (k = 13, k1 = 2)

Table 1: Values of k and k_1

k	2	1	2	2	13	2
k_1	2	2	1	13	2	12

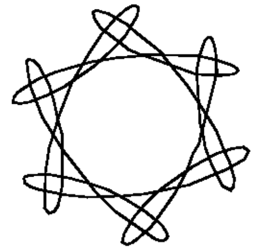


Fig. 11: Trajectory (k = 2, k1 = 12)

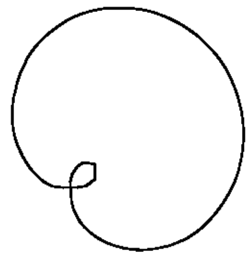


Fig. 12: Trajectory (k = 2, k1 = 2)

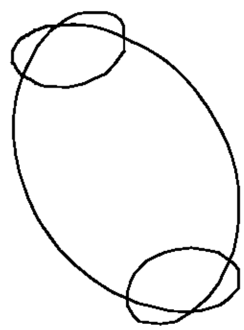


Fig. 13: Trajectory (k = 1, k1 = 2)

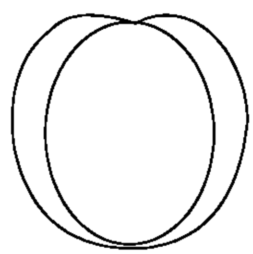


Fig. 14: Trajectory (k = 2,k1 = 1)

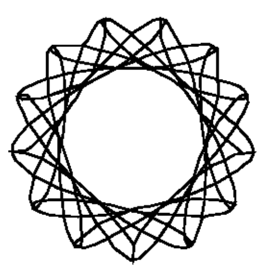


Fig. 15: Trajectory (k = 2, k1 = 13)

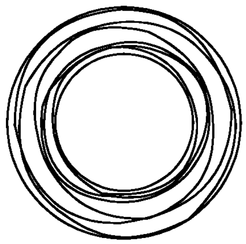


Fig. 16: Trajectory (k = 13, k1 = 2)

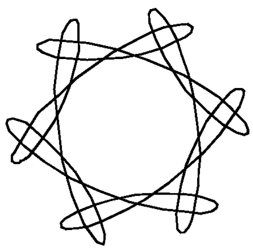


Fig. 17: Trajectory (k = 2, k1 = 12)

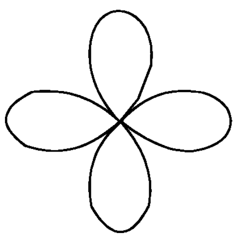


Fig. 18: Trajectory (k = 1, k1 = 2)

matrix of parameters

$$P = \begin{bmatrix} 1 & -u_k \\ -u_k & 1 \end{bmatrix}$$

is adopted. The initial conditions are chosen equal to those adopted in the previous configuration while keeping the values of k and k_1 of the Table 1. The Fig. 18-24 illustrate the obtained results. In Fig. 21 only a part of trajectory has been illustrated (with a reduced number of points).

Results of the fourth algorithm configuration: The implementation of the fourth algorithm structure (12), in which the matrix of parameters

$$P = \begin{bmatrix} 1 & +u_k \\ +u_k & 1 \end{bmatrix},$$

is illustrated by the Fig. 25-28. The initial conditions are chosen equal to those adopted in the previous configuration while keeping the values of k and k_1 of the Table 1. In Fig. 28 only a part of trajectory has been illustrated (with a reduced number of points).

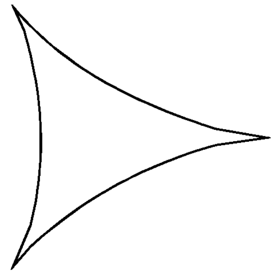


Fig. 19: Trajectory (k = 2, k1 = 2)

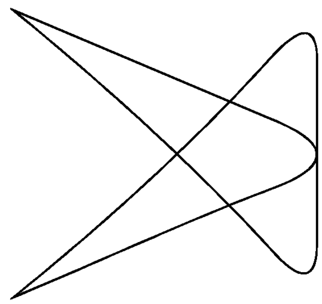


Fig. 20: Trajectory (k = 2, k1 = 1)

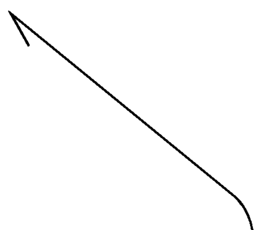


Fig. 21: Trajectory (k = 2, k1 = 1)

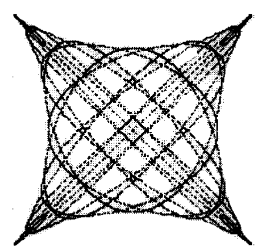


Fig. 22: Trajectory (k = 13, k1 = 2)

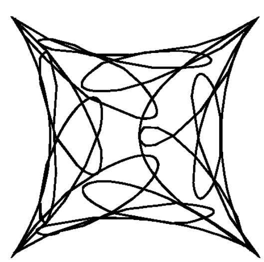


Fig. 23: Trajectory (k = 2, k1 = 13)

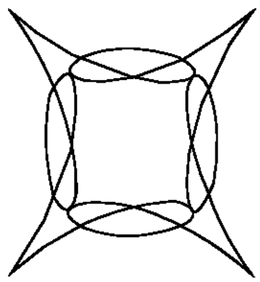


Fig. 24: Trajectory (k = 2, k1 = 12)

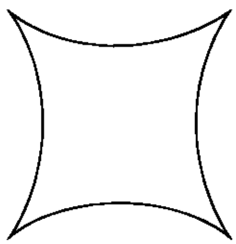


Fig. 25: Trajectory (k = 1, k1 = 2)

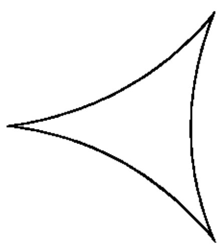


Fig. 26: Trajectory (k = 2, k1 = 2)

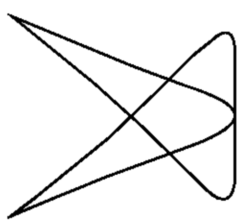


Fig. 27: Trajectory (k = 2,k1 = 1)

Influence of a parameters vector addition on the trajectories form: While modifying the equations system (13) by the addition of a constant vector w_1 such as $w_1^T = [0 \quad \lambda]$ and while adopting the first configuration of the algorithm (12):

$$\begin{pmatrix} u_k \\ v_k \end{pmatrix} = \begin{pmatrix} \alpha_1 & \beta_1 \\ -\beta_1 & \alpha_1 \end{pmatrix} \begin{pmatrix} u_{k-1} \\ v_{k-1} \end{pmatrix} + \begin{pmatrix} 0 \\ \lambda \end{pmatrix} \qquad (14)$$

One could get the trajectories forms, Fig. 29-34. The implementation has been done with the following values of initial conditions and of the vector w_1:

$u_0 = 1$, $v_0 = 1$, $x_0 = 1$, $y_0 = 1$ and . $w_1^T = [0 \quad -0.75]$

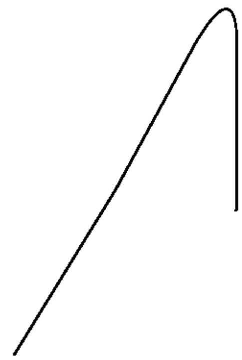


Fig. 28: Trajectory (k = 2, k1 = 1)

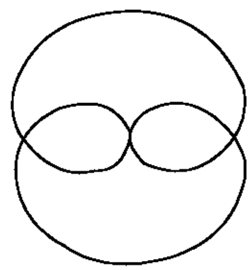


Fig. 29: Trajectory (k = 2, k1 = 1)

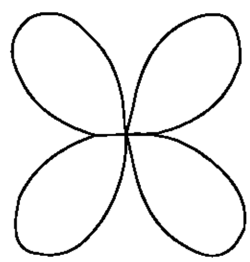


Fig. 30: Trajectory (k = 1, k1 = 2)

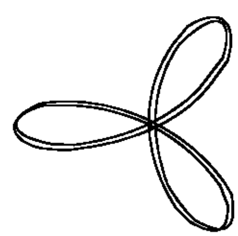


Fig. 31: Trajectory (k = 1, k1 = 3)

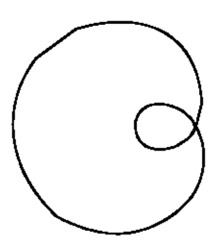


Fig. 32: Trajectoire (k = 3, k1 = 1)

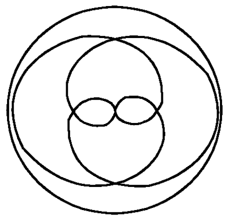


Fig. 33: Trajectoire (k = 4, k1 = 1)

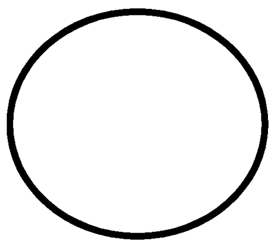


Fig. 34: Trajectoire(k = 1, k1 = 1)

ROBOT MOTION CONTROL WITHOUT OBSTACLES

The specific problem that will be studied, in this section, is the application of the algorithm (11) to the robot motion control from an initial position to a final position on the plane surface. This point-to-point motion should be done so that optimizing the energy spent in the motion. Indeed, the motion can be effected along an alternated circular form trajectory on both sides of a straight path joining these two points without obstacles collision with a minimal energy spending. The Fig. 35-36 illustrate our approach.

Analytic study of control: The segment of right joining the points is divided in identical elementary sections, determined by the angle of robot curvature (Fig. 37 and 38). The extremities of these sections represent the intersection points of the robot symmetrical elementary paths with the considered segment. These instants of meeting allow the algorithm of control to impose following a symmetrical elementary path in order to reach the arrival point. The optimisation of the energy spent during the motion in the absence of obstacles is limited by the robot curvature constraint. The alternated circle arcs must have a minimal curvature radius. The paths scheduling for mobile robot is given in the reference (Fraichard and Scheuer, 2004).

Let us consider a point M of the right D, defined by the equation:

$$y = ax+b$$

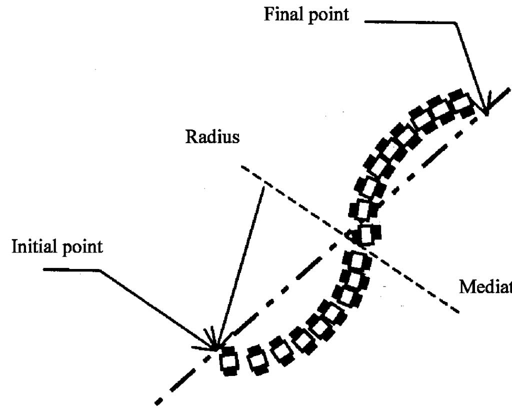


Fig. 35: Trajectory of alternated circular form

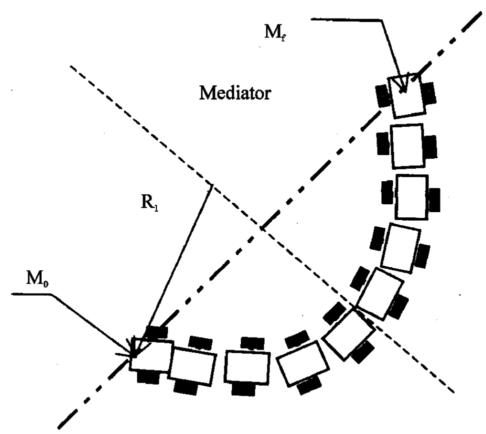


Fig. 36a: Trajectory (form of circle arc (R_1))

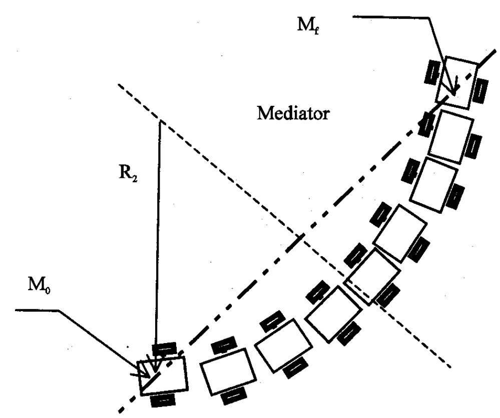


Fig. 36b: Trajectory (form of circle arc (R_2>R_1))

The right D is determined by the two points M_0 and M_1, this implies:

$$a = \frac{y_f - y_0}{x_f - x_0} \text{ et } b = \frac{y_0 x_f - y_f x_0}{x_f - x_0}$$

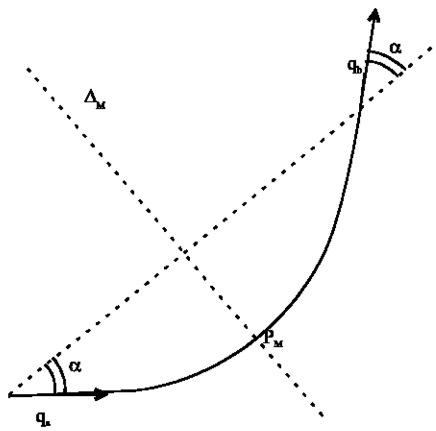


Fig. 37: Angle of the robot curvature

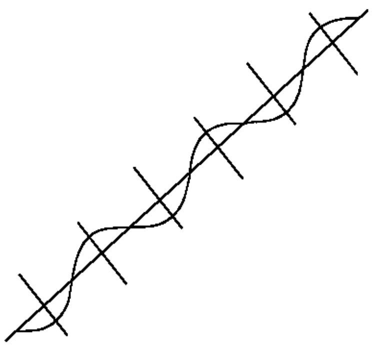


Fig. 38: Division of the segment in n elementary section

One notes by M_e, the middle of segment of right (M_0M_f) and by Δ_M, its mediator (Fig. 37). All point M such as $M\epsilon\Delta$ represents a circle center of radius $Mm_i = MM_f$. More that one moves away of M_e in the positive sense more one optimises the energy consumed by the robot during its motion:

one can write:

$$M_0\begin{pmatrix} x_0 \\ y_0 \end{pmatrix} \text{ initial point}$$

$$M_f\begin{pmatrix} x_f \\ y_f \end{pmatrix} \text{final point } \begin{cases} x_f = x_n \\ y_f = y_n \end{cases}$$

$$\begin{pmatrix} x_n \\ y_n \end{pmatrix} = \begin{pmatrix} \cos(2\pi\theta) & -\sin(2\pi\theta) \\ \sin(2\pi\theta) & \cos(2\pi\theta) \end{pmatrix}\begin{pmatrix} x_{n-1} \\ y_n - 1 \end{pmatrix}$$

$$\begin{pmatrix} x_{n-1} \\ y_{n-1} \end{pmatrix} = \begin{pmatrix} \cos(2\pi\theta) & -\sin(2\pi\theta) \\ \sin(2\pi\theta) & \cos(2\pi\theta) \end{pmatrix}\begin{pmatrix} x_{n-2} \\ y_{n-2} \end{pmatrix}$$

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} \cos(2\pi\theta) & -\sin(2\pi\theta) \\ \sin(2\pi\theta) & \cos(2\pi\theta) \end{pmatrix}\begin{pmatrix} x_0 \\ y_0 \end{pmatrix}$$

avec $\alpha = \cos(2\pi\theta)$ et $\beta=\sin(2\pi\theta)$

$$\begin{pmatrix} x_n \\ y_n \end{pmatrix} = \begin{pmatrix} \cos(2\pi n\theta) & -\sin(2\pi n\theta) \\ \sin(2\pi n\theta) & \cos(2\pi n\theta) \end{pmatrix}\begin{pmatrix} x_0 \\ y_0 \end{pmatrix}$$

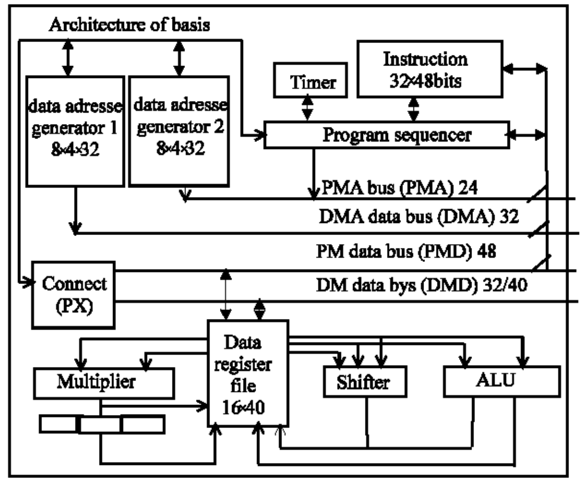
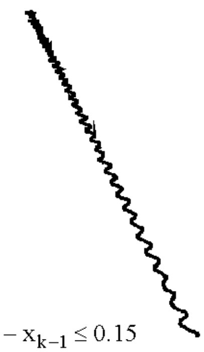
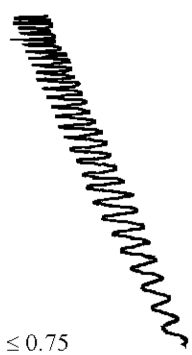


Fig. 39: Implantation of the control algorithm



(a): $-0.15 < x_k - x_{k-1} \leq 0.15$



(b): $-0.75 < x_k - x_{k-1} \leq 0.75$

Fig. 40: Robot motion from a point M_0 toward a point M_f,

IMPLANTATION OF THE ROBOT CONTROL ALGORITHM ON DSP

In the goal to validate our theoretical survey concerning the robot motion from an initial point M_0 toward a final point M_f; one has implemented the algorithm on a kit to basis of a Digital Signal Processor DSP: 21065L SHARC (Super Harvard architecture Computer).

Presentation of the platform: As all DSPs, the Sharc is a processor constructs to accelerate the execution of the algorithms dedicated to the numeric treatment of the signal. It possesses an architecture of Harvard type. In such architecture (Fig. 39), one distinguishes the Program Memory PM and the Data Memory DM. Each of these memories is accessible by a different bus. It is therefore possible to reach the instructions simultaneously and to the data, that in first approximation, permits to multiply by two the global speed of the machine, while increasing the flux of informations. The Sharc manages its space of addresses while using the Data Addresses Generators: DAG1 and DAG2. The first generator of addresses (DAG1) permits to control the 32 bytes of the bus DM (Dated Memory), whereas DAG2 manages the addresses of the PM bus (Program Memory) coded on 24 bytes. Every DAG is constituted of four types of registers: Registers of Index (I), of Modify (M), of Bases (B) and of Lengths (L). The registers are numbered from 0 to 7; those of DAG2 of 8 to 15. The I register contains the address that will be displayed on the bus of the corresponding memory and M the value of the increment. The new address will be calculated automatically while adding to the value of the I register, the content of the register M according to the formula:

$$I \leq ((I + M - B)\%L) + B \tag{15}$$

Implantation of the control algorithm: The validation of the control algorithm on DSP ADSP 21065L has been done for paths in form of circle arcs. The control is led in the absence of obstacles. One sees that it is possible to choose some trajectories in the form of circle arcs of a determined curvature radius and that can be fixed by the geometric robot constraints and by the optimisation ones in energy expense. The Fig. 40a,b have illustrate the implementation results. The survey of the robot motion control will be the subject of a future contribution; and this for trajectories of simple or complex forms and in presence of obstacles.

CONCLUSIONS

One presented, in this study, the paths trajectories scheduling for the non holonomic mobile robots. In this purpose, one studied a dynamical numeric type strategy, based on the dynamic systems theory and that permits to generate any canonic path form (simple or complex). A dynamic system is a system which is governed by a recursive algorithm. Besides, it is bound closely to the initial conditions responsible of its behavior and to the number of adopted iterations. The survey, presented in this study, aims the trajectories scheduling where the considered motions planning is a continuous sequence in the robot space. The affected trajectories are of closed or open type. One was interested, in the last section, to the analytic survey of the robot motion control from an initial given point toward an intended final point and this while proposing an approach of energy optimisation spent in its motion following alternated circle arcs on both sides of the right joining these two points. Every circle arc of minimal curvature radius has been chosen so that the carrier circle center is a point belonging to the mediator of the right portion joining its two extremities. In addition, the curvature radius is computed along the tangential angle link to this circle arc with the one that, immediately, follows it.

In perspectives to this study we would consider this continuity of the curvature along a path for a mobile robot in presence of obstacles. The motion control will be studied for trajectory paths containing one or many obstacles.

REFERENCES

Fraichard, T. and A. Scheuer, 2004. From reeds and shepp's to continuous-curvature paths. IEEE Trans. Robotics, 20: 1025-1036.

Hui, N.B., V. Mahendar and D.K. Pratihar, 2006. Time-optimal, collision-free navigation of a car like mobile robot using neuro-fuzzy approaches. Fuzzy Sets and Systems, 157: 2171-2204.

Kumar, V., M. Zefran and J. Ostrowski, 1997. Motion Planning and control of Robots. Handbook of Industrial Robotics, J. Willey and Sons, October.

Laumond, J.P., 1987. Finding collision-free smooth trajectories for a non holonomic mobile robot. In Proceeding of the International Joint Conferance on Artificial Intelligence, Milan (IT), August, pp: 1120-1123.

Liang, T.C., J. Liu, G. Hung and Y. Chang, 2005. Practical and flexible path planning for car-like mobile robot using maximal-curvature cubic spiral. Robotics and Autonomous System, 52: 312-335.

Reeds, J.A. and L.A. Shepp, 1990. Optimal paths for a car that goes both forword and backwords. Pacific J. Math., 145: 367-393.

Scheuer, A. and C.H. Laugier, 1998. Planning sub-optimal and continuous-curvature paths for car-like robot. In: Proceeding of the 1998 IEEE/RSJ. International Conference on Intelligent Robot and System, Victoria, BC, Canada, October, pp: 25-31.