15th Triennial World Congress, Barcelona, Spain

HIGH LEVEL PATH COORDINATION FOR MULTIPLE VEHICLES AND RECIPROCAL ROOT LOCUS

Aldo Balestrino, Alberto Landi

Dipartimento di Sistemi Elettrici e Automazione (DSEA)
Università di Pisa, Via Diotisalvi, 2, I-56126 Pisa, Italy
Email: landi@dsea.unipi.it

Abstract: This paper presents a method based on the reciprocal root locus (RRL) approach for solving the multiple vehicle path coordination problem. A high level planning assigns the tasks to each vehicle, specifying each mission, inside a structured workspace with known obstacles. The problem is solved in two steps: first the given workspace is mapped onto a semicircle, using the theory of conformal mappings, then the RRL method is applied. Examples involving multiple vehicles illustrate the proposed technique and its effectiveness. *Copyright © 2002 IFAC*

Keywords: Multiple mobile vehicles, Root locus, High-level path coordination.

1. INTRODUCTION

Several authors have investigated the multiple vehicle path coordination problem (Latombe, 1991). The problem at hand considers n vehicles sharing the same planar environment among stationary obstacles: a high-level algorithm computes the path planning, so that each vehicle can reach its own target without collisions. The problem addressed in this paper is a classic one in robot motion planning, nevertheless it can be easily assimilated to mobile vehicles in assembly plants, or automated factories, or military tasks, or airplane taxing. Most of the literature involved is devoted to a centralised approach (e.g., Kant and Zucker, 1986; Erdmann and Lozano Perez, 1987), but also a decentralised approach as in Tournassoud (1986) or in Lumelsky and Harinarayan (1997) has been widely approached.

The solution proposed in this paper is based on the theory of conformal transformations that can be applied to map the given workspace in a simpler geometric setting. Assuming for the sake of simplicity the workspace as a semicircle, the path planning can be easily solved using the reciprocal root locus (RRL) technique, a design technique used for solving the discrete-time quadratic regulator problem (Kailath, 1980; Jacquot, 1981).

RRL method does not take into account the presence of obstacles during the path coordination phase: obstacle avoidance can be formally included in a pre-processing step, inside the conformal mapping, so that a virtual obstacle-free workspace can be considered. Such pre-processing phase can be computationally too heavy, in case of a complex environment with many obstacles. Therefore computational burden can be simplified using a post-processing step for obstacle avoidance: some heuristic rules can be added to simplify the mapping when the workspace (including obstacles) looks very complex. In the paper some examples of complex environments are reported and simple heuristic rules are applied. Nevertheless it must be put into evidence that the primary target of this paper is the idea of using reciprocal root locus for planning and that different and more refined methods for obstacle avoidance could be easily introduced in the post-processing phase (e.g. Lumelsky and Stepanov, 1987).

The path-coordination algorithm proposed in this paper is based on the RRL method: it solves in a quick and simple way the multiple mobile vehicle problem. The properties of such algorithm guarantee the computed paths cannot intersect. Since path intersections are prevented, in a well-known environment without mobile obstacles the vehicles do not need complex planning capabilities. On line computations, data exchanges and high level local capabilities are not as necessary as in many decentralized systems proposed in literature. Unfortunately the non-intersecting property has disadvantages in all cases requiring intersecting solutions: for example two vehicles switching positions in a narrow corridor. Such cases are discussed in the final remarks of the paper.

2. STATEMENT OF THE PROBLEM

Given the available vehicles, the tasks to be executed, and a description of an environment, at the highest planning level a first problem is the assignment of tasks, the second one is a path coordination. If all vehicles are equipped with sensors and with some computing capability, then each vehicle can manage local planning and, partially, obstacle avoidance. Hence at the highest level the planner is only needed for specifying missions.

A high-level path coordination is considered under the following assumptions:

a) mobile vehicles are equally equipped (i.e., any vehicle can execute the same task);
b) the environment is bounded;
c) the obstacles and the working places (targets) are fixed and known.

The path coordination problem must be solved avoiding collisions between vehicles and between vehicles and obstacles.

Remark: at a glance, the assumption a) seems an undesirable simplification. Such hypothesis is less restrictive when we have no priority in choosing which vehicle must reach a given target. In the case of specialised vehicles we can easily select the vehicle oriented to the task, if all vehicles are starting from the same place. Moreover assumption a) can be relaxed, if necessary, by considering only a vehicle and a target turn and turn about and replying the procedure for each pair of vehicle and target. This individual (or partially grouped) approach can produce global intersecting paths: the solution is still admissible only if we allow vehicles to intersect with appropriate priority (Kant and Zucker, 1989).

Assumption c) implies that if changes happen in an environment, an on-line replanning of vehicles must be quickly performed.

The planner must address the following goals:

1. choosing the target for each vehicle: if the vehicles are more than the targets, choosing which vehicle has to operate
2. finding the paths for the mobile vehicles: simultaneous movements must be allowed and vehicle collisions avoided
3. obstacle avoidance.

3. WORKSPACE REPRESENTATION AND CONFORMAL MAPPINGS

If a solution is known for a given workspace and for a particular arrangement of vehicles and tasks, i.e. if we know every path joining a vehicle and the final location without collisions, a conformal transformation will lead to another admissible solution (Kober, 1952).

Of course the conformal mapping will modify the boundary and the planned paths. A theorem by Riemann (Cohn, 1980) states that there exists a conformal mapping between the domain internal to a simply-connected region R and the interior of a unit circle; moreover this mapping is unique if we prescribe that the centre C and a direction pointing outward from the centre are mapped respectively into a point of R and a direction going through it. More specifically we know that a polygon with rectilinear edges can be conformally mapped onto the unit circle, e.g. by using the Schwartz-Christoffel formula.

Hence if the workspace is described by a polygon we can use the conformal transformation to map the workspace onto the interior of the unit disk; by solving the path planning problem for multiple mobile vehicles in this setting, we can recover the right solution by means of the inverse conformal transformation. In order to get a practical algorithm it is more convenient to assume the domain R to be mapped onto the unit circle as composed by the workspace, for instance a polygon, and a copy of the workspace obtained by a reflection of it with respect to an edge AB of the polygon. In this way we can use the principle of symmetry by Schwartz-Riemann, so that the workspace is mapped onto the semicircle with positive imaginary part. Moreover a continuos function f(z) olomorph in R and on its boundary, with f(z) real if z belongs to the edge AB, can be analytically prolonged also on the copy of R obtained by reflection with respect to the edge AB so that f(z) assumes complex conjugate values in points symmetrically arranged with respect to the edge AB.

In the following we assume from the beginning that the workspace is represented as the upper half of the unit circle; such hypothesis allows to solve the path planning for multiple mobile vehicles by drawing a reciprocal root locus.

Note that such assumption on the workspace representation can still include all cases in which the workspace is described by a polygon and the conformal mapping can be applied. If the obstacles are a-priori known, their mapping inside the semicircle can be computed with an *off-line* procedure.

It must be remarked that the direct and inverse mapping between the workspace R (including obstacles) and the upper half of the unit circle is often a hard task to be performed by on-line computations.

4. THE RECIPROCAL ROOT LOCUS APPROACH

In order to get a suitable algorithm for path planning of multiple vehicles in a given workspace a potential function (Khatib, 1986; Rimon and Koditschek; 1992) can be used. If we require some minimal smoothness property it is quite natural to assume as a potential function a harmonic one i.e. a function satisfying almost everywhere the Laplace equation. In our setting the potential function is chosen as a meromorphic function f(z), i.e. a function characterised only by its poles and zeroes. The poles are representative of the vehicles and the zeroes of the targets, hence we can build up the function

$$f(z) = T(z)/R(z) \qquad (1)$$

with

$$T(z) = (z-t_1)(z-t_1^*)(z-t_2)(z-t_2^*)..(z-t_m)(z-t_m^*) \qquad (2)$$

$$R(z) = (z-p_1)(z-p_1^*)(z-p_2)(z-p_2^*)..(z-p_n)(z-p_n^*) \qquad (3)$$

where t_i and p_k represent respectively the target and the vehicle locations and t_i^* and p_k^* are their complex conjugate singularities.
The proposed method is based on the reciprocal root locus (RRL) technique, an effective and widely used design technique of the linear discrete-time quadratic regulator problem. Let us briefly sketch the standard usage of the RRL technique in the case of a single input- single output linear system

$$\begin{aligned} x(k+1) &= Ax(k)+bu(k); \qquad x(0)=x_0 \\ y(k) &= c^T x(k). \end{aligned} \qquad (4)$$

The problem is to choose the control u such that the performance index

$$J_N = \sum_{i=0}^{N-1} q y^2(i) + r \sum_{i=0}^{N-1} u^2(i) \qquad (5)$$

is minimised, where x is the vector of the states, q is a weighting factor, r is a cost parameter and T is the transpose operator. It can be shown (see Jacquot, 1981 for a detailed analysis) that the optimum solution is

$$u_{opt}(i) = -k(i)x(i) \qquad (6)$$

where k_i is a gain depending on the solution of a discrete Riccati equation. It can be shown that if the pair (A,b) is controllable, the pair (c,A) is detectable and $q=c^Tc$, then k(i) tends toward a constant value k. The optimal steady-state gain k satisfies

$$\det(zI - A + bk) = \prod_{i=1}^{n} (z - z_i) \qquad (7)$$

where the z_i's are the n stable roots (i.e. lying within the unit circle) of the 2n-th degree polynomial equation

$$1+(q/r)G(z^{-1})G(z)=0 \qquad (8)$$

where $G=c(zI-A)^{-1}b$.
Equation (8) is the so-called RRL: it represents the discrete version of the well-known symmetric root locus in the continuous-time case. It can be used in selecting the optimal locations of the n stable roots for a given q/r gain by using a root-locus technique and in such a way the optimal steady-state gain k is obtained without solving the discrete Riccati equation. Note that only well-known simple and timesaving algorithms (typical of the root locus) are required.
The proposed procedure, applied to the planning for multiple vehicles is named*: Root Locus Path Coordination* (RLPC). Except on a few situations (possible points of multiple roots), the different paths do not intersect, hence every vehicle can proceed at the most suitable speed, and vehicles can move simultaneously, without any risk of collision. Some properties of the root locus are usefully exploited: e.g. different branches of a root locus do not intersect. This property looks a perfect solution for the planner: assimilating the vehicles to the poles and the targets to the zeros of a root locus, a path selection avoiding collisions between vehicles is automatically achieved. The various paths must be constrained inside the border of the workspace: this feature can be easily guaranteed by the RLPC. In fact the border is represented on the complex plane by the segment [-1, 1] on the real axis and by the upper unit semi-circumference; poles and zeros will be placed inside this region respectively in correspondence with the positions of vehicle and targets. A well-known property of root locus guarantees the branches remain inside this region if the singularities are symmetric w.r.t. the real axis. From (1) and (8) the characteristic equation of the reciprocal RLPC is:

$$R(z)\,R(1/z) + KT(z)\,T(1/z) = 0 \qquad (9)$$

where K is a parameter to be referred as the gain.
Consider the case in which the vehicles are more than the targets. Let n and m be the degrees of R(z) and T(z), with n>m and n-m=2h. In the root locus there will be 2h “infinity zeros”, and 2h zeros in (0,0) given by the reciprocal terms, R(1/z) and T(1/z). In the upper complex half-plane there will be h poles reaching the origin, belonging to the upper unit half-disc and corresponding to the inactive vehicles, able to reach a sort of “parking” in the origin. Note that this “parking” point can be moved to a different position, by introducing zeros in the most suitable places, instead of the origin, without affecting the efficiency of the algorithm.
The case in which the targets are more than the vehicles is analogous to the previous one by interchanging the roles of poles and zeroes. The algorithm will assign to the exceeding targets some

ideal vehicles at the parking point; the corresponding tasks can be accounted for in a following planning.

5. HEURISTIC RULES FOR OBSTACLE AVOIDANCE

A major difficulty arises whenever conformal mappings from workspace to circle are required on line: to overcome this problem some heuristic rules are added in order to obtain feasible paths. If paths for the various vehicles are traced without accounting for obstacles, hence collisions can take place. In these cases few heuristic rules can be added to solve the problem in a practical and simple way. Obstacles have to be modelled: the proposed solution considers augmented obstacles inscribed in a circle or in an ellipse. The planner must avoid any intersection between the paths and the circles. For example the obstacles, in a stationary case, may be grown by the shape of the vehicles. In such hypothesis the proposed algorithm follows the steps:

- finding the paths by computing the RLPC in a free workspace (without obstacles);
- the path from P_1 to P_2 (forbidden area) is removed;
- connecting P_1 to P_2 with the shortest circumference arc outside the "forbidden area".

In such a way, any planning inside a "forbidden area" is avoided and obstacle avoidance can be easily performed. The *a-priori* knowledge of P_2 is necessary to find the better way (clockwise or counter clockwise) to trace the path from P_1 to P_2.

Obstacles could have shapes that do not suggest surrounding them with a circumference (for instance a very long obstacle). In order to circumvent the problem, surroundings could be defined by using ellipses instead of circles; this solution does not modify the main structure of the algorithm.

Critical cases can occur leading to a conflict among vehicles:

i. different vehicles are required to follow close paths with an interference risk;
ii. more than one path arrive at a "forbidden area" and the heuristic rule may suggest a common circumference arc;
iii. two (or more) mobile vehicles interchange their position in a workspace made of narrow corridors.

The above cases can be solved only if a suitable priority in the planning is considered. Prioritisation rules have been widely studied in literature (e.g., Erdmann and Lozano Perez, 1987).

If the vehicles are equipped with sensors, a simple local planning can solve most of the problems. Such re-planning can be executed locally by repeating the RLPC procedure if sufficient computation power is available and if a mapping of the vehicle neighbourhood is known.

6. EXAMPLES AND RESULTS

Several cases of generated trajectories have been considered for testing the RLPC technique. In the following figures, the initial location of each vehicle is represented by a cross, the targets by circles and the planned path by a full line.

In Fig.1 a trivial planning is described: two mobile vehicles are shown in a free environment. In the figure only the branches of the RRL corresponding to the stable solution interior of the upper half of the unit circle are drawn, i.e. the RLPC.

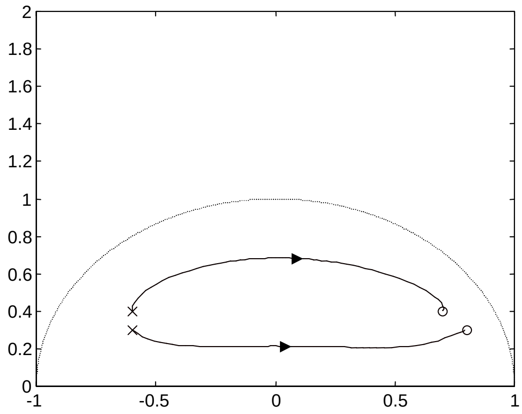

Fig. 1. Two mobile vehicles in a free environment

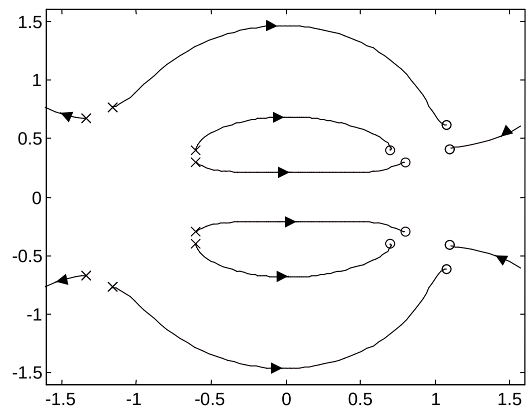

Fig.2. Complete RRL in the case of Figure 1

In figure 2 the complete RRL of Fig.1 is illustrated, to make clearer the method. It includes the singularities corresponding to the initial locations and targets of the vehicles inside the upper half of the unit circle, the complex conjugate singularities and the reciprocal singularities due to the RRL method. Due to symmetry properties of the root locus, the branches corresponding to the vehicles always remain inside the upper half of the unit circle and different branches do not intersect.

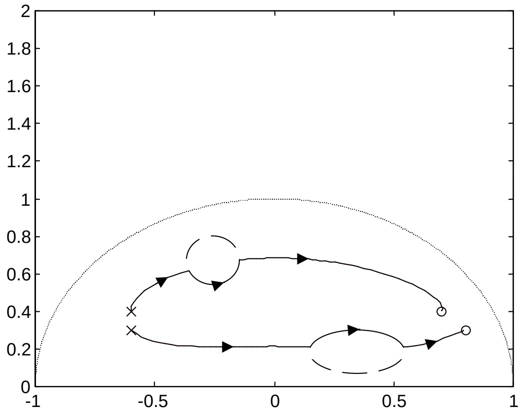

Fig. 3. Two mobile vehicles with obstacle avoidance

In Fig.3 the case of Fig.1 is repeated in the presence of two augmented obstacles (circle-shaped and ellipse-shaped) interfering with the computed solution: the heuristic rules of section 5 are applied to the planner, resulting in an extremely fast first-search planning.

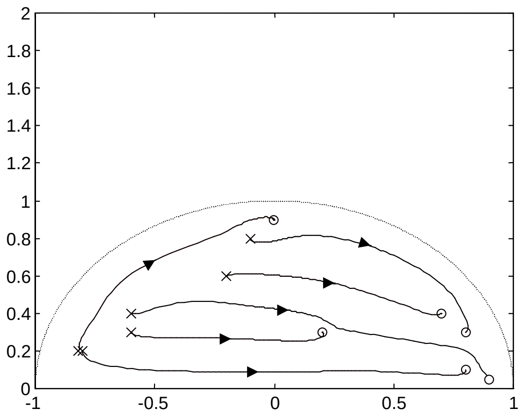

Fig. 4. Six mobile vehicles

In Fig.4 a RLPC in a free environment is shown with six mobile vehicles (two of them are starting from the same 'parking' area): it can be noted the automatic distribution of the targets with not intersecting paths. In Fig.5 a case of parking at the origin of a group of vehicles is shown.

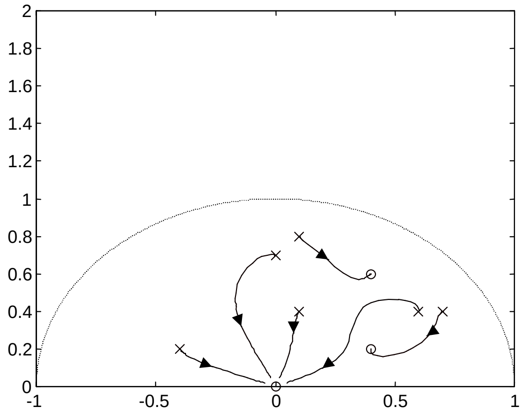

Fig. 5. Parking for a group of vehicles

Criticism: in the case of a cross poles-zeros symmetry the RLPC does not achieve optimal results, as shown in Fig.6. The first search algorithm leads to unsatisfactory paths, although the task is still satisfied. These particular cases may be treated by introducing additional heuristic procedures, e.g. the RLPC can be replanned introducing a different mapping of the workspace: of course with a loss of simplicity and increasing the computation time. Anyhow the case presented in Fig.6 is the worst one: it is too simple to put in evidence any advantage of the RLPC and a high level planning is unnecessary (straight lines are the optimal solution). Different symmetries do not affect the planning quality.

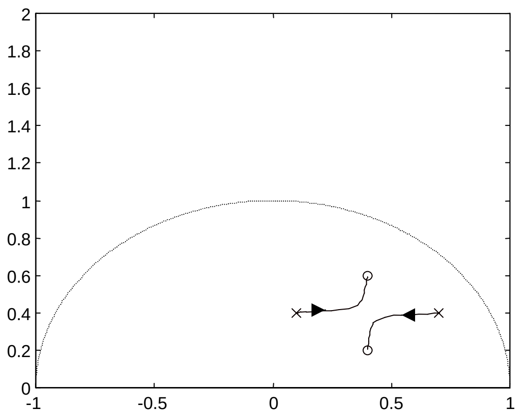

Fig. 6. Cross pole-zero symmetry

In Fig.7 a complex case is illustrated: there are 8 vehicles, 2 of them parking at the origin, 2 with a cross pole-zero symmetry and in the presence of 3 obstacles. Simulation results have been obtained by modifying the standard root locus routine of the Matlab software package. The upper unit half-disc is presented on the screen, along with the whole planning, with the assignment of targets to vehicles and the paths to follow. It must be highlighted that the solution is extremely fast, although the RLPC is implemented with a general-purpose software. In the case of a dedicated implementation (e.g. rewriting the routines in a C code) such time can be dramatically reduced (Landi and Paoletti, 1995). Moreover the algorithm efficiency is extremely high, being based on the well-known and widely checked algorithms solving the root locus problems.

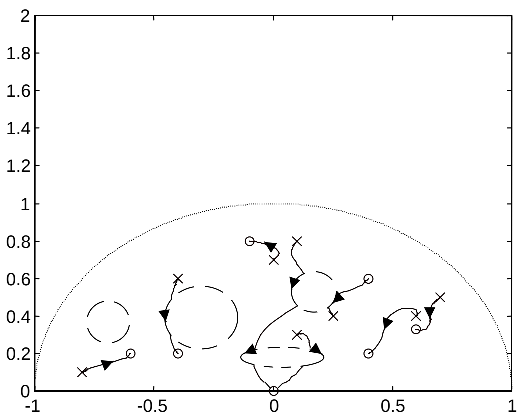

Fig. 7. Eight mobile vehicles in a complex environment with obstacle avoidance

7. CONCLUDING REMARKS

Main advantages of the approach proposed are:

- the algorithm solves in a simple and practical way the multiple vehicle path coordination problem at the highest control level
- powerful computing hardware is not required and vehicles do not need a complex equipment for their motion
- after a first high level planning, vehicles do not need a complex navigation system but they only must follow the assigned path till the target is reached
- the planner is extremely simple. It always solves the problem with a first search without collisions
- heuristic rules can be added in order to cope with obstacle avoidance, without any further computational burden.

Drawbacks of the approach are:

- it cannot be applied in the case of vehicles interchanging their positions;
- the case of different vehicles required to follow close paths (or the same paths if they are intersecting a common obstacle) with an interference risk can be solved considering the velocities of the single vehicles and priority rules in the case of interference;
- moving, or time-varying obstacles require multiple applications of the RLPC algorithm.

Such critical cases are revealed, but they cannot be solved using the RLPC algorithm by itself. Unfortunately they require an on-line local replanning of each vehicle, equipped with sensor systems, losing the simplicity of the solution proposed. The approach proposed appear general: in case of a practical application to a realistic scenario it must be reconsidered after a careful analysis of possible configurations. Based on the RLPC high-level path coordination, simple rules (e.g., priority rules) can be added.

REFERENCES

Cohn, H. (1980). *Conformal mapping on Riemann surfaces.* Dover Publications, New York.

Erdmann M. and T. Lozano Perez (1987). On multiple moving objects. *Algorithmica*, **2**, 477-521.

Kailath, T. (1980). *Linear systems*. Prentice-Hall, Inc., Englewood Cliffs, New York.

Kant, K., and S.W. Zucker (1986). Toward efficient trajectory planning: path velocity decomposition. *International Journal of Robotics Research*, **5**, 72-89.

Khatib, O. (1986). Real time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research,* **5,** 90-98.

Kober, H. (1952). *Dictionary of conformal representations*. Dover Publications, New York.

Jacquot, R.G. (1981). *Modern digital control systems*. M.Dekker Inc., New York.

Landi, A., Paoletti, F. (1995). Reciprocal root locus approach to path planning for multiple mobile robots. *Proceedings INRIA/IEEE Symposium on Emerging Technologies and Factory Automation, EFTA'95*, October 1995, Paris, 345-352.

Latombe, J.C. (1991). *Robot motion planning*. Norwell, MA: Kluwer Academic Publishers.

Lumelsky, V.J., and A.A. Stepanov (1987). Path planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape. *Algorithmica*, **2**, 403-430.

Lumelsky, V.J., and K.R. Harinarayan (1997). Decentralized motion planning for multiple mobile robots: the cocktail party model. *Autonomous Robot*, **4**, 121-135.

Rimon, E., and D.E. Koditschek (1992). Exact robot navigation using artificial potential functions. *IEEE Trans. on Robotics and Automation*, **5**, 501-518.

Tournassoud, P. (1986). A strategy for obstacle avoidance and its application to multi-robot systems. *Proceedings IEEE Int. Conf. On Robotics and Automation*, 1224-1229.