*XI Reunión de Trabajo en Procesamiento de la Información y Control, 21 al 23 de septiembre de 2005*

# Route planning for vehicle autonomous navigation, based on geometrical regions.
# Part II: Multiple approach points

L. M. Di Matteo[1], A. C. Mangone[2], M. L. Muzzio[3], C. Verrastro[4]

[1] *Artificial Intelligence Group, Univ. Tecnológica Nacional Fac. Regional Bs. As., Buenos Aires, Argentina.*
*Leandro.DiMatteo@ieee.org*
[2] *Artificial Intelligence Group, Univ. Tecnológica Nacional Fac. Regional Bs. As., Buenos Aires, Argentina.*
*sigma1@ciudad.com.ar*
[3] *Universidad Argentina de la Empresa, Facultad de Ingeniería, Buenos Aires, Argentina.*
*maximiliano@termoacustica.com.ar*
[4] *Artificial Intelligence Group, Univ. Tecnológica Nacional Fac. Regional Bs. As., Buenos Aires, Argentina.*
*Centro Atómico Ezeiza, Comisión Nacional de Energía Atómica, Buenos Aires, Argentina.*
*cverra@cae.cnea.gov.ar*

***Abstract*** **-- This paper presents an algorithm to generate approach waypoints in autonomous mobile navigation, in order to take a vehicle to a target point, arriving to it with a specified entry angle.**

**Here, we present an improved technique in the approximation phase for the route planner presented in Part I.**

**This algorithm can be added to the one presented in the Part I to have a solid technique to address the problem of trajectories and approximation planning in static or dynamic environments for mobile vehicles with considerable inertial factors.**

**We model the environment using geometrical regions and it is not grid based, that is why we use algebraic expressions to calculate waypoints. It is assumed that the environment model is known all the time.**

***Keywords*** **-- Robotics, autonomous navigation, path planning, optimum routing, mobile vehicle navigation.**

## I. INTRODUCTION

Path planning problems have brought several study cases in the last decade, but since the beginning of this new decade, many of these problems are focused in solving systems adapted to highly dynamical environments.

In a recent work (Di Matteo et al, 2004b) we proposed an algorithm for route planning applied to robotics in highly dynamical environments, it was described as a simple algorithm that generates a trajectory from an initial to a goal position, avoiding obstacles, where the goal point moves all the time and also the obstacles, and where the processing time must be as short as possible.

As the planning was based in geometrical regions, it presented an excellent time response for real time systems.

After experimental results, the implementation accomplished successfully the path planning, but we found a not desired consequence in the goal point approximation due the big inertia of the mobile robot considered.

In the Part I of this work, we define that the robot should reach the goal point with a specified entry angle, for that reason it was added an approximation point forming the desired angle with the goal point. When the robot runs to the approach point, if the angle formed by it and the goal point compared with the entry angle is enough small, that is to say, less than 30 degrees approximately, the goal point is reached with the right entry angle, but if these angle differences are bigger it is noted a deviation of the desired entry angle in the final point regarding the real entry angle.

In this work, we propose a method to improve the performance of the goal point tracking, minimizing the entry angle error.

To do this, multiple approach points are used, instead of only one, it lets the mobile to reach N previous points before arriving to the goal point. The amount of approach points N is determined for each application.

This algorithm was implemented in the UTN FRBA robot soccer team (Di Matteo et al, 2004a) for the 'II Argentine Robot Soccer Championship', and it showed to be very effective.

## II. OBJECTIVES

The main task of this technique is to improve the performance of the goal point tracking, minimizing the entry angle error, in the algorithm presented in the previous work.

It replaces the use of a single approach point by multiples points disposed in a determined way, in order to correct gradually the entry angle error.

Current and target position must be given, as well as the environment knowledge.

## III. ENVIRONMENT MODELLING

As mentioned in Part I, we work with an environment without a cell oriented structure. Therefore, we use algebraic expressions to model objects and trajectories.

We assume a two dimensional world, and where each obstacle is a square with a known side length.

In case those obstacles are irregular, not squares, the environment must be preprocessed to put it in the squared form.

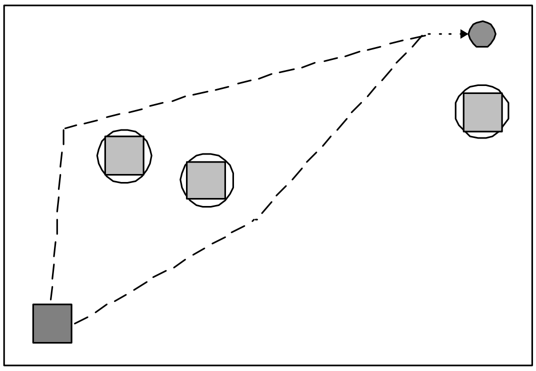

Figure 1.

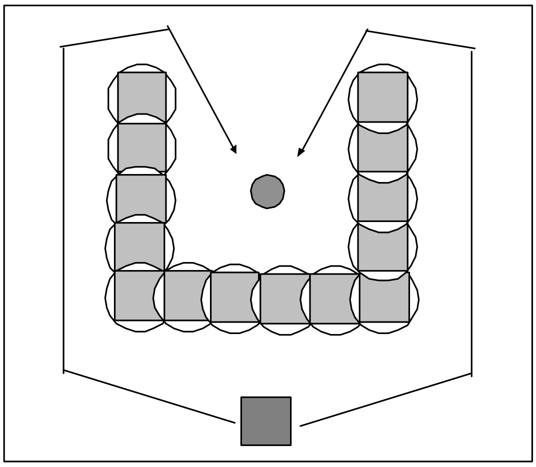

Figure 2.

Once we have a modularized workplace, for each obstacle block the mass center must be calculated, that is to say, the center of the square in a squared obstacle.

Each obstacle block is represented with a circumference, Eq. 1, where the center is the mass center of the object, and its radius is predefined by the user, Fig. 1 and Fig. 2.

$$(X - Xc)^2 + (Y - Yc)^2 = Rrobot^2 \qquad (1)$$

It is intended to configure the obstacle radius a little greater than obstacle diagonal, with the aim of taking into account the mobile robot dimensions, in order to avoid lateral collision between the robot and obstacles.

## IV. GENERAL DESCRIPTION

As input parameters, it must be given the following: current mobile position, goal point position, desired entry angle and the environment model.

Multiples approach points are arranged, the amount of them are defined depending on the application. These points are previous waypoints that the mobile must go trough before arriving to the goal point.

These waypoints are mounted over the line that passes trough the target point and which has the angle equal to the entry angle. Obviously, these are located in the space closed between the mobile current position and the goal position, as shows Fig. 3.

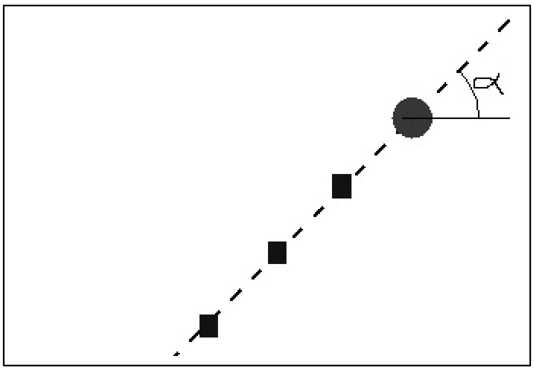

Figure 3.

In order to correct de entry angle error gradually, the distance between points is not linear, the distance is increased exponentially, Eq. 2, where k factor should be selected depending on the dynamical characteristics of the mobile and the environment, if the environment moves faster or the mobile robot has a bigger inertial factor, k must be higher. The variable i refers to the number of approach point, in which i=1 refers to the closest waypoint to the goal point, and i=N refers to the last point. These distances are calculated just one time, in the first run, in order to optimize the algorithm speed, and are stored in an array.

$$D_i = e^{i.k} \qquad (2)$$

To obtain a better time response when arriving to the final point, we had to leave the idea of exact way points to pass to define bands, where each band is denoted by the subtraction of the circumferences centered in the goal point which radius are given by the approach point i and i+1, Fig. 4.

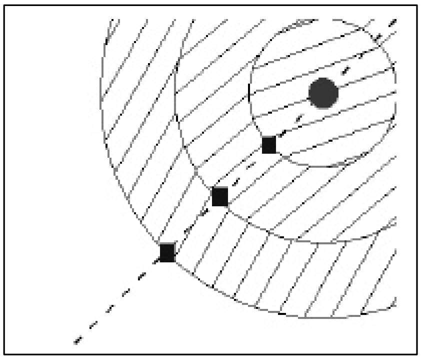

Figure 4.

Now, the mobile robot must travel to the waypoints form N, N-1, N-2, …., to 1, but with the modification that if the mobile steps on the next region, that is to say, it is located nearer the goal point than the waypoint is, the current waypoint to reach is changed by the following, i+1, without reaching the programmed waypoint i. It is illustrated in Fig. 5.

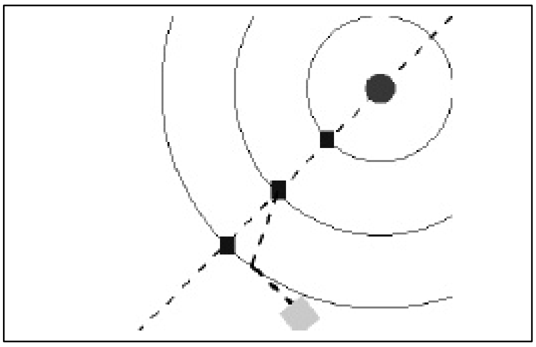

Figure 5.

Other improvement introduced, is the addition of error bands placed on each approach point. These new regions provide a smoother path in the robot trip.

Concerning to the robot navigation, the same concept mentioned in the paragraph above is applied.

As shown in Fig. 6, each error band is centered on each approach point Eq. 3, and the respective radiuses are function of de distance of the point to the target point Ec. 1003.

$$R_{ErrBand}^{i}{}^{2} = \left(X - X^{i}_{AppPo\text{int}}\right)^2 + \left(Y - Y^{i}_{AppPo\text{int}}\right)^2 \quad (3)$$

$$R_{ErrBand}^{i} = K.i \quad (4)$$

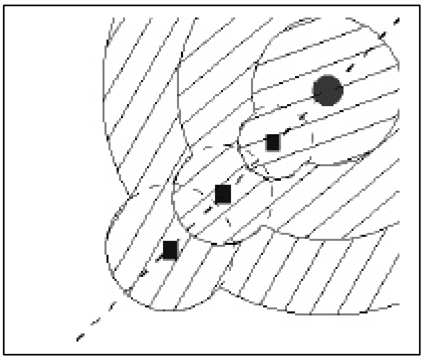

Figure 6.

In Fig. 6, it is also visible how geometrical regions are conformed, intersecting approximation bands with error bands.

As final remark, the approximation algorithm is the following, expressed in pseudocode:

```
//////////////////////////////////////////////////////////////////////////
    ApproachPoint = N

    DO
        NAVIGATE TO ApproachPoint
        IF CurrentPosition IS inside RegionN THEN
            ApproachPoint = ApproachPoint - 1
    LOOP UNTIL ApproachPoint = 0

    NAVIGATE TO GoalPoint
//////////////////////////////////////////////////////////////////////////
```

It is illustrated in the following figure, Fig. 7:

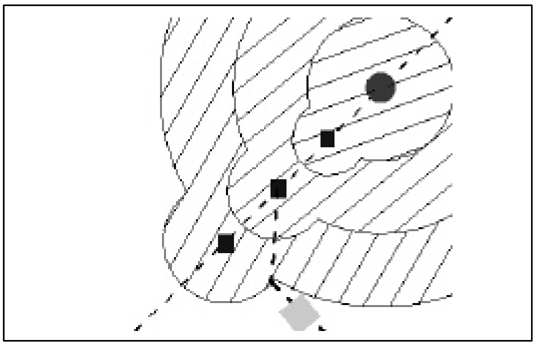

Figure 7.

## VIII. EXPERIMENTAL RESULTS

To test our work we use a robot soccer simulation platform, more precisely, the simulator v1.4 used by FIRA organization in its tournaments.

A short code in C++ language was written down to test the algorithm, all data was logged to a text file. Some results are shown in Fig. 8 to 11.

The dark gray box represents the mobile robot, in this case we use a robot soccer player, and the goal point is the ball. In all cases, the desired entry angle is 45 degrees.

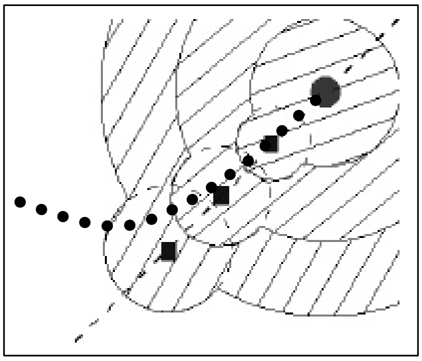

Figure 8.

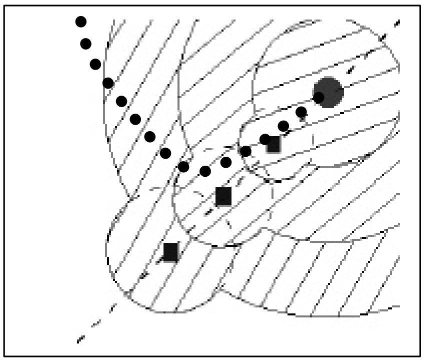

Figure 9.

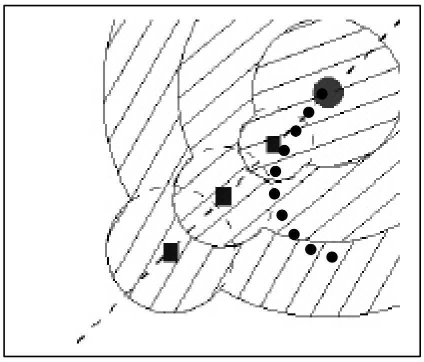

Figure 10.

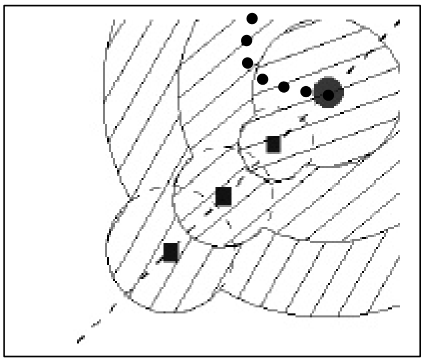

Figure 11.

## IX. CONCLUSIONS

In experimental tests, the improvement in the approximation phase was evident. With this new addition, the approximation to the goal point is smoother and the error in the entry angle has decrease considerably.

This paper address the problem of approximation planning to a target point in static or dynamic environments when the mobile vehicle has a high inertial behavior, where the inertial factor can not be ignored.

Therefore, if we run this algorithm in conjunction with the one presented in the Part I of the article, we have a solid technique to address the problem of trajectories and approximation planning in static or dynamic environments for mobile vehicles with considerable inertial factors.

As seen in the experimental results, the algorithm proposed achieves its objective successfully; the simulation demonstrated that it fits to static and dynamic environments.

## REFERENCES

Di Matteo, L., Mangone, A., Muzzio, M., "Equipo de Fútbol de Robots de la UTN FRBA", II Campeonato Argentino de Fútbol de Robots, CAFR 2004, Tandil, Bs. As., Argentina (2004a).

Di Matteo, L., Mangone, A., Muzzio, M., Verrastro C., "Route Planning for Vehicle Autonomous Navigation. Part I: Single Approach Point", V Jornadas Iberoamericanas de Robótica", Santa Cruz de la Sierra, Bolivia. (2004b).

Khatib, O. "Real-time obstacle avoidance for manipulators and mobile robots" Proceedings IEEE-ICRA, St. Louis MO, 500-505 (1985).

Latombe, J. C. "Robot Motion Planning", Kluwer Academic Pub. Boston (1991).

Orqueda O. and Agamennoni O. "Motion Planning and control of Autonomous robots – I: Generalized Potential Field Functions" X RPIC Proceedings, San Nicolás, Bs. As., 541-546 (2003).

Patiño D. and Carelli R. "Adaptive Critic Design-Based Optimal Control For Mobile Robots Navigation", X RPIC Proceedings, San Nicolás, Bs. As., 503-507 (2003).