Predictive and Multi-rate Sensor-Based Planning under Uncertainty

Marta C. Mora and Josep Tornero, *Member, IEEE*

***Abstract*— In this paper, a general formulation of a predictive and multi-rate reactive planning method for intelligent vehicles is introduced. The method tackles path planning and trajectory planning for intelligent vehicles in dynamic environments with uncertainty, where the kinodynamic vehicle constraints are also taken into account. It is based on the Potential Field Projection method (PFP), which combines the classical Potential Fields method with the Multi-rate Kalman filter (MR-KF) estimation. PFP takes into account the future object trajectories and their associated uncertainties, what makes it different from other look-ahead approaches. Here, a new potential field is included in the Lagrange-Euler formulation in a natural way, accounting for the vehicle dynamics. The resulting accelerations are translated into control inputs that are considered in the estimation process. This leads to the generation of a local trajectory in real time that fully meets the constraints imposed by the kinematic and dynamic models of the intelligent vehicle. The properties of the method are demonstrated by simulation with Matlab© and C++ applications. Very good performance and execution times are achieved, even in challenging situations. In a scenario with 100 obstacles a local trajectory is obtained in less than 1s, which is suitable for real time applications.**

***Index Terms*— Dynamics, intelligent vehicles, Kalman filter, potential fields, sensor-based planning, trajectory prediction, uncertainty.**

I. Introduction

SENSOR-based planning for intelligent vehicles is an active research field. Although the planning problem was defined twenty years ago [1], there is still work to do in order to obtain the full autonomy for a vehicle in uncertain environments.

Current trends are concerned with computing motion in unknown, dynamic and cluttered environments [2]. For this purpose, intelligent vehicles have to gather, process and use sensor information in real-time (RT). Under this conditions path planning is frequently carried out by reactive methods, such as Potential Fields (PF) [2, 3], VFH* [4], Nearness Diagram [5], Curvature-Velocity method (CVM) [6] and the Dynamic Window approach (DW) [7, 8], among others [2]. These methods compute collision-free local paths adapting to changes in the environment in RT. They are usually combined with a global planning technique to obtain a global collision-free path [2]. Still, they do not always behave well in complex situations, especially when high velocities are involved. This is due to any of these reasons: i) vehicle kinodynamic features are not considered; ii) objects' movements are ignored; iii) uncertainty in measurements is not taken into account.

Some methods, such as CVM and DW, consider the vehicle dynamics in the path generation while the majority transfers the problem to the control field [9-11]. However, they do not include motion prediction in their computation.

Conversely, look-ahead approaches consider the objects' movement. In [4] different robot trajectories are evaluated using the A* method and a cost function. However, neither the obstacles' trajectories nor uncertainty are taken into account. In this sense, [12] use an evolutionary PF based on a relative threat coefficient, that computes relative velocities of target, robot and obstacles. Also [13] considers future obstacles' trajectories but none of these last methods account for uncertainty or vehicle dynamics.

Regarding to uncertainty in measurements, few methods [14-16] include it in the planning process. In [14] a probabilistic description of the environment is built based on the analogy of an electric current flowing in a conductor. Then, a harmonic potential field is used to compute the path. Nonetheless, the resulting holonomic path may not be feasible for non-holonomic vehicles as dynamics is not considered. In [15] an algorithm to solve the stochastic shortest path problem is proposed that, by adjusting suboptimal bounds on solutions and reusing previous searches, provides good performance. However, nothing is said about the dynamic properties of the robot and the resulting path may be unfeasible for a wheeled robot. [16] presents a sensor-based planning approach that accounts for uncertainty, motion prediction and computes a smooth navigation function using E*. Still, the vehicle dynamics is not considered.

In [17, 18] we presented a multi-rate (MR) predictive PF method for local planning that takes all the previous aspects into account, unlike other methods [2-16] that only consider some of them. The Potential Field Projection method (PFP) combines the PF method [3] with the multi-rate Kalman filter (MR-KF) estimation [19, 20], thus considering present and future vehicle and obstacles locations within the planning strategy. Moreover, the original concept of the PF method [3] for motion generation was reintroduced, in the sense that artificial forces determine the vehicle acceleration.

In this paper, a new formulation of the method is presented using the Lagrange-Euler (LE) equations for solving the

Manuscript received July 15, 2013 (date submitted for review).

M. C. Mora is with the Mechanical Engineering and Construction Department, Universitat Jaume I, Spain (corresponding author to provide phone: +34 964 72 81 26; fax: +34 964 72 81 06; e-mail: mmora@uji.es).

J. Tornero is with the Design and Manufacture Institute, Universidad Politécnica de Valencia, Spain (e-mail: jtornero@idf.upv.es).

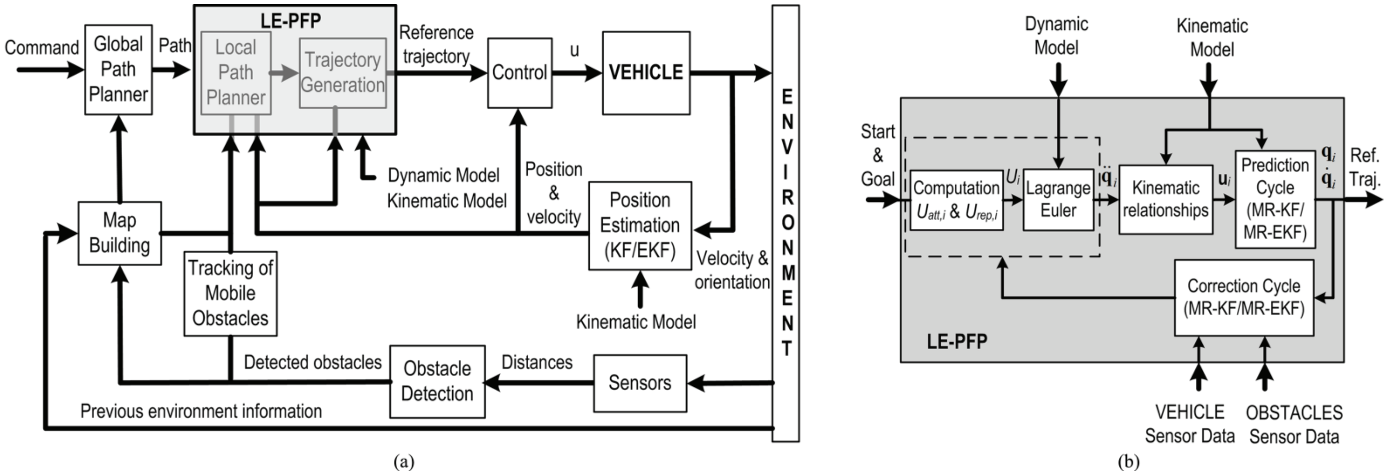

Fig. 1. (a) Generic deliberative software architecture for an intelligent vehicle, (b) Block diagram of the sensor-based method LE-PFP.

forward dynamic problem. This formulation, named LE-PFP, provides a holistic framework that can be used for any kind of vehicle, with any kinematic and dynamic model. Also a key issue is tackled: the merging of the position uncertainty and the object volume for its consideration in the planning method.

In order to show the capabilities of the methodology, a complete analysis for a unicycle vehicle with a dynamic model is developed as well as different simulation applications.

The paper is organized as follows: section II introduces the path planning/trajectory generation problem within a deliberative architecture, section III describes the PFP method; section IV develops the integration of the LE formulation with the PFP method; section V presents a non-linear example; sections VI and VII show simulation results in different and challenging environments (Matlab & C++) while section VIII gives conclusions and future work.

II. Sensor-Based Planning Based On LE-PFP

The LE-PFP method proposed here can be used in a typical deliberative architecture, as the one shown in Fig. 1, where all the software modules required for autonomous navigation have been included. A global planning module computes an initial collision-free path to be followed by the vehicle, in the form of a set of intermediate goals $\mathbf{q}_{goal}$. Although a global planning technique computes a geometric free-collision path that guarantees global convergence, this is only true in known and static environments. If the scenario is uncertain and dynamic, this path has to be modified in RT to tackle uncertain situations. The success of the global trajectory will depend on the type of environment (cluttered or clear, partially known or unknown, slow or fast obstacles, etc.), the type of sensors used (accuracy, acquisition rate, etc.), the suitability of the models employed, etc.

The on-line path modification is usually carried out by local methods as new sensor information is gathered. These methods provide a local path as a sequence of configurations $\mathbf{q}$, which has to be converted into a reference trajectory in order to consider the vehicle kinematic constraints.

One of the advantages of the LE-PFP method with respect to other local approaches is that it combines local path planning and trajectory generation within a single module, which performs both tasks simultaneously thereby saving computational time (Fig. 1(a)). Details regarding its internal structure are shown in Fig. 1(b) and explained next.

On the one hand, the system can be considered affected by an artificial and conservative potential field U_i generated by the goal and the obstacles at the present (i=0) and future (i>0) instants of time (PFP method). U_i modifies the vehicle motion equations (and resulting accelerations $\ddot{\mathbf{q}}_i$) if included in the LE formulation, allowing a natural integration of the vehicle dynamics with the navigation using PF methods.

On the other hand, we can highlight the presence of a feedback loop in a similar fashion as in a typical control structure. This feedback, along with the fact that a force derived from a potential field has the structure of a controller (see (17)), involves the extension of the classical philosophy for robot control to the field of planning. Indeed, the attractive force, on the structural aspect of its equations, can be considered as a P-control (proportional) on the vehicle position, in its simplest form, or as a PD-control (proportional-derivative), if the attractive force is added to a friction force, thus also considering velocities [18].

This implies that control inputs $\mathbf{u}_i$ are derived from the accelerations $\ddot{\mathbf{q}}_i$ by kinematic relationships. These inputs are used in the trajectory prediction carried out by MR-KFs.

As a result, a reference trajectory for the control is obtained, as it generates a path with a kinematic profile. This trajectory fully meets the kinodynamic vehicle constraints, for both holonomic and non-holonomic vehicles.

The reference trajectory is updated through the correction cycle of the MR-KF/EKF whenever sensor measurements are available for the vehicle and/or the obstacles. It is a continuous replanning that quickly adapts to changes in the environment.

III. Potential Field Projection Method (PFP)

Some of the most popular reactive methods are based on the artificial Potential Field method [2, 3]. These techniques compute a local path in an iterative way, re-computing it in RT as new sensor information is gathered. The path is determined assuming that obstacles generate repulsive forces while the goal generates attractive forces acting on the vehicle. These forces are derived from artificial potential functions.

The method is fast, mathematically elegant and simple.

In the context of pedestrian modeling, social force models arise to model pedestrian motion in a similar way to that of PF methods [21-23]. The underlying concept is the same: the pedestrian behavior is driven by attractive and repulsive potential fields. The social force represents the motivation to act (or move) of the pedestrian [21].

Nonetheless, path planning techniques based on PF typically generate holonomic paths and, therefore, don't produce feasible solutions for vehicles. Besides, traditional PF techniques consider deterministic locations for the vehicle and the obstacles in a particular instant of time. This is not a realistic assumption due to the uncertainties in measurements and models. This inherent uncertainty implies that path planning must aim at maximizing the chances of reaching a goal without collision. In this sense, the objects future trajectories should be taken into account to achieve safer planning, mainly in high speed applications.

Finally, PF methods usually run at a determined *frame period, $T_d=mT$*, different from the *control period T*, because of the diversity on sensor rates. This means that between two consecutive measurements there is an interval where the vehicle navigates with past information.

The Potential Field Projection method considers questions not addressed in other PF methods: i) position uncertainties; ii) different execution periods of control-estimation (T) and planning algorithms ($T_d=mT$; $m \in Z^+$) due to diverse sensor rates; iii) future objects locations during the *prediction horizon* ($T_h=N \cdot T$; $N \in Z^+$) based on their kinematic models.

PFP generates, on every execution and from the available sensor data, a set of forces that guide the vehicle to a set of positions in the present (i=0) and future (i=1,..,N) instants, differing from other PF methods [2, 3] that do not perform trajectory prediction.

Pedestrian obstacles are modeled with second-order linear kinematic models, as in (1), where $\mathbf{x} \in \Re^{nx1}$ is the state vector, $\boldsymbol{u} \in \Re^{mx1}$ is the control input vector, $\boldsymbol{z} \in R^{px1}$ is the measurement vector, $\mathbf{A} \in \Re^{nxn}$, $\mathbf{B} \in \Re^{nxm}$, $\mathbf{C} \in \Re^{pxn}$ and $\mathbf{G} \in \Re^{nxp}$ are the state space matrices and $\mathbf{w} \in \Re^{px1}$, $\boldsymbol{v} \in \Re^{px1}$ are the process and measurement noise, respectively.

$$\mathbf{x}_{i+1} = \mathbf{A} \cdot \mathbf{x}_i + \mathbf{B} \cdot \mathbf{u}_i + \mathbf{G} \cdot \mathbf{w}_i \qquad \mathbf{z}_i = \mathbf{C} \cdot \mathbf{x}_i + \mathbf{v}_i \tag{1}$$

Vehicle obstacles and intelligent vehicles (IV) are modeled with second-order non-linear models, as in (2), that consider their kinematic restrictions, being **f** and **h** non-linear functions of the state and the input, respectively.

$$\mathbf{x}_{i+1} = \mathbf{f}\left(\mathbf{x}_i, \mathbf{u}_i, \mathbf{w}_i\right) \qquad \mathbf{z}_i = \mathbf{h}\left(\mathbf{x}_i\right) + \mathbf{v}_i \tag{2}$$

Using these models, a set of predicted future positions and uncertainties ($\hat{\mathbf{x}}_i, \mathbf{P}_i$) is obtained running the prediction cycle of the MR-KF/EKF (3) or (4) for every object in the environment during the prediction horizon T_h. The absence of measurements in future instants of time (i>0) is modeled with *delta* functions [17, 19, 20], a MR technique for time-varying models that modifies the Kalman gain indicating the presence ($\boldsymbol{\Delta}_i$=**I**) or absence ($\boldsymbol{\Delta}_i$=**0**) of measurements in every estimation instant. The choice of the filter to be used depends on the kinematic model associated to the object. For linear models MR-KF (3) will be used while MR-EKF (4) is appropriate for non linear models.

$$\begin{array}{c|l} \textit{Prediction} & \hat{\mathbf{x}}_{i/i-1} = \mathbf{A} \cdot \hat{\mathbf{x}}_{i-1/i-1} + \mathbf{B} \cdot \mathbf{u}_{i-1} \\ \textit{cycle} & \mathbf{P}_{i/i-1} = \mathbf{A} \cdot \mathbf{P}_{i-1/i-1} \cdot \mathbf{A}^T + \mathbf{G}_{i-1} \cdot \mathbf{Q}_{i-1} \cdot \mathbf{G}_{i-1}{}^{\mathrm{T}} \\ & \mathbf{K}_i = \mathbf{P}_{i/i-1} \cdot \mathbf{C}^{\mathrm{T}} \cdot \left[\mathbf{C} \cdot \mathbf{P}_{i/i-1} \cdot \mathbf{C}^{\mathrm{T}} + \mathbf{R}\right]^{-1} \cdot \boldsymbol{\Delta}_i \\ \textit{Correction} & \hat{\mathbf{x}}_{i/i} = \hat{\mathbf{x}}_{i/i-1} + \mathbf{K}_i \cdot \left[\mathbf{z}_i - \mathbf{C} \cdot \hat{\mathbf{x}}_{i/i-1}\right] \\ \textit{cycle} & \mathbf{P}_{i/i} = \mathbf{P}_{i/i-1} - \mathbf{K}_i \cdot \mathbf{C} \cdot \mathbf{P}_{i/i-1} \end{array} \tag{3}$$

$$\begin{array}{c|l} \textit{Prediction} & \hat{\mathbf{x}}_{i/i-1} = \mathbf{f}\left(\hat{\mathbf{x}}_{i-1/i-1}, \mathbf{u}_{i-1}\right) \\ \textit{cycle} & \mathbf{P}_{i/i-1} = \mathbf{F}_{i-1} \cdot \mathbf{P}_{i-1/i-1} \cdot \mathbf{F}_{i-1}^{\mathrm{T}} + \mathbf{W}_{i-1} \cdot \mathbf{Q}_{i-1} \cdot \mathbf{W}_{i-1}^{\mathrm{T}} \\ & \mathbf{K}_i = \mathbf{P}_{i/i-1} \cdot \mathbf{H}_i^{\mathrm{T}} \cdot \left[\mathbf{H}_i \cdot \mathbf{P}_{i/i-1} \cdot \mathbf{H}_i^{\mathrm{T}} + \mathbf{R}_i\right]^{-1} \cdot \boldsymbol{\Delta}_i \\ \textit{Correction} & \hat{\mathbf{x}}_{i/i} = \hat{\mathbf{x}}_{i/i-1} + \mathbf{K}_i \cdot \left[\mathbf{z}_i - h_i\left(\hat{\mathbf{x}}_{i/i-1}\right)\right] \\ \textit{cycle} & \mathbf{P}_{i/i} = \mathbf{P}_{i/i-1} - \mathbf{K}_i \cdot \mathbf{H}_i \cdot \mathbf{P}_{i/i-1} \end{array} \tag{4}$$

$$\mathbf{F}_{i-1} = \frac{\partial \mathbf{f}\left(\hat{\mathbf{x}}_{i-1/i-1}, \mathbf{u}_{i-1}\right)}{\partial \mathbf{x}} \qquad \mathbf{W}_{i-1} = \frac{\partial \mathbf{f}\left(\hat{\mathbf{x}}_{i-1/i-1}, \mathbf{u}_{i-1}\right)}{\partial \mathbf{w}} \qquad \mathbf{H}_i = \frac{\partial \mathbf{h}\left(\hat{\mathbf{x}}_{i/i-1}\right)}{\partial \mathbf{x}} \tag{5}$$

In the previous equations $\hat{\mathbf{x}} \in \Re^{nx1}$ is the state estimation vector, $\mathbf{P} \in \Re^{nxn}$ is the error estimation variance matrix, $\mathbf{K} \in \Re^{nxp}$ is the Kalman gain and $\mathbf{R} \in \Re^{pxp}$ and $\mathbf{Q} \in \Re^{pxp}$ are the measurement noise and the process noise covariance matrices, respectively, that are considered constant matrices. **R** is computed based on sensor noise and transmission errors while **Q** is evaluated experimentally using ground truth measurements. $\mathbf{F} \in \Re^{nxn}$ and $\mathbf{W} \in \Re^{nxp}$ are the Jacobian of $\mathbf{f}(\cdot)$ with respect to the state and to the process noise, respectively, while $\mathbf{H} \in \Re^{pxn}$ is the Jacobian of $\mathbf{h}(\cdot)$ with respect to the state. These matrices derive from the linearization of the non-linear model, as stated in (5).

Remark that data regarding obstacles (type, intention and motion direction) have to be provided by a tracking module (see Fig. 1), which processes the sensor information. This module has to account for the dynamics of the obstacles. In the case of pedestrians, the social force model could be used as well as recently proposed local motion models such as human motion prediction model [24] or linear trajectory avoidance [25]. In the case of vehicles, tracking algorithms like [26] could be used. In this sense, the LE-PFP method makes a short-term prediction considering that, during T_h, motion direction is not going to change significantly. These little changes can be modeled by the noise terms in (1) and (2).

A. A Geometric Model for the Uncertainty

The predicted trajectories are described as a sequence of states and uncertainty matrices ($\hat{\mathbf{x}}_i, \mathbf{P}_i$) for each object. The evolution of **P** depends on the availability of measurements and the noise matrices **Q** and **R**. These matrices are highly related to the characteristics of the hardware used and have to be determined experimentally. Typical sensors of common use in mobile robotics are a) exteroceptive: ultrasonic sensors, laser rangefinders, cameras, GPS etc., b) proprioceptive: encoders, accelerometers, gyroscopes, etc., (see [27-29] for in-depth descriptions and error computation). The specific sensors used have to be chosen attending to the environment and the velocities of vehicle and obstacles.

As the KF estimates the state $\mathbf{x}_i$ through a Gaussian distribution, the confidence region of the estimate $\hat{\mathbf{x}}_i$ can be described by a g-sigma ellipsoid [29] given in (6), where $\mathbf{X}_i$ is each of the points of the ellipsoid surface in the i-th prediction instant. Thus, each $(\hat{\mathbf{x}}_i, \mathbf{P}_i)$ tuple can be geometrically modeled by an ellipsoid. In path planning applications where RT computation is required, only the estimated positions and related uncertainties are used in the ellipsoid generation. Thus, the ellipsoid can be considered a forbidden navigation area.

$$(\mathbf{X}_i - \hat{\mathbf{x}}_i)^{\mathrm{T}} \cdot \mathbf{P}_i^{-1} \cdot (\mathbf{X}_i - \hat{\mathbf{x}}_i) \leq g^2 \tag{6}$$

The constant value g indicates the probability P for the mobile object to be inside the uncertainty ellipsoid and is obtained using the expression (7) for the Gaussian distribution [30, 31], which results in the simpler equation (8) for n=2.

$$P = P\left\{(\mathbf{X}_i - \hat{\mathbf{x}}_i)^{\mathrm{T}} \cdot \mathbf{P}_i^{-1} \cdot (\mathbf{X}_i - \hat{\mathbf{x}}_i) \leq g^2\right\} = \frac{2}{2^{n/2}\,\Gamma\left(\frac{n}{2}\right)} \int_0^g t^{n-1} e^{-t/2} dt \tag{7}$$

$$P = 1 - e^{-\frac{g^2}{2}} \tag{8}$$

For safety reasons, a probability of 98.9% has been selected (g=3) for planning, being (9) the new ellipsoid equation.

$$(\mathbf{X}_i - \hat{\mathbf{x}}_i)^{\mathrm{T}} \cdot \mathbf{P}_i^{-1} \cdot (\mathbf{X}_i - \hat{\mathbf{x}}_i) \leq 9 \tag{9}$$

As g is a scalar value, an equivalent equation can be obtained considering the increment within the matrix $\mathbf{P}_i$, i.e., we can substitute $\mathbf{P}_i(98.9\%)=9\cdot\mathbf{P}_i$ obtaining:

$$(\mathbf{X}_i - \hat{\mathbf{x}}_i)^{\mathrm{T}} \cdot (9 \cdot \mathbf{P}_i)^{-1} \cdot (\mathbf{X}_i - \hat{\mathbf{x}}_i) \leq 1 \tag{10}$$

Note that the more probability required the bigger the ellipsoid considered for planning. Any other desired P value implies analogous expressions: $\mathbf{P}_i(P\%)=g^2\cdot\mathbf{P}_i$.

In order to make planning feasible, a maximum limiting uncertainty matrix $\mathbf{P}_i$ should be established. The optimum value $\mathbf{P}_{optim}$ can be obtained off-line with (3) in the case of a linear system or in a previous test stage with (4) in the case of a non-linear system. This value is directly related to the maximum ellipsoid radius. Typically, this maximum radius must be one order of magnitude lower than the dimension of the setting where the vehicle is moving. Applying this idea the *estimation quality index* defined in [17] can be computed as in (11), being β a user adjustable distribution scaling factor.

$$J = \exp\left(-\left(trace(\mathbf{P}) - trace\left(\mathbf{P}_{optim}\right)\right)^2 / \beta\right) \tag{11}$$

The J value is a measure of the prediction quality: a J value near 1 means that the estimation corresponds to a possible optimum while near 0 means the absence of measurements due to the multi-rate nature of the process or because the object is hidden. If J decreases under a threshold limit J_{min}, (fixed by the user according to hardware features), it means a too long absence of object measurements and an excessive increase of the predicted trajectory uncertainty. This situation occurs when the obstacle has moved away from the vehicle and is not sensed any more. Thus, it is removed from the list of obstacles. In the case of the vehicle, planning becomes unfeasible and the vehicle stops until new information is gathered which enables restarting the navigation process.

B. Considering the Object Volume

The above mathematical description does not include the object volume. We consider the object volume in a very similar manner to the construction of C-obstacles.

First, we express the object volume in an analogous formulation to that of the ellipsoid, that is, we model objects by means of spherical volumes introduced in [32]. This modeling technique represents an object as the convex hull of a finite set of spheres, which is the spherical extension of a polytope (*s-tope*). Given a set of spheres S={s$_0$, s$_1$,..., s$_n$}, an *s-tope* of order n is given by (12), where each sphere s$_i$ is called *spherical vertex* and is described using two parameters s=($\mathbf{c}$; r), the center $\mathbf{c}\in\Re^3$ and the radius $r\in\Re$. The parameter λ identifies the intermediate spheres between two spherical vertices in the *s-tope* and the order n identifies the number of spherical vertices composing the *s-tope*. The geometric modeller in [33] can be used for *s-tope* model generation in the case of complex geometries for the vehicle and obstacles. Otherwise, simple enveloping spheres can be used.

$$\mathrm{S}_{0n} = \left\{ \mathrm{s}:\ \mathrm{s} = \sum_{i=0}^{n} \lambda_i \cdot \mathrm{s}_i,\ \mathrm{s}_i \in \mathrm{S},\ \lambda_i \geq 0,\ \sum_{i=0}^{n} \lambda_i = 1 \right\} \tag{12}$$

A sphere is an ellipsoid with identical radii r in every direction and can be described with a matrix equation $\mathbf{X}^{\mathrm{T}}\cdot\boldsymbol{\Lambda}\cdot\mathbf{X}\leq 1$ similar to (6), where $\boldsymbol{\Lambda}=diag(1/r^2)$.

Second, we merge the object volume with the uncertainty ellipsoid. Let M be an uncertainty ellipsoid defined by (6) in $\Re^n$ obtained from the trajectory prediction of a moving object $O\in\Re^n$, where $\mathbf{P}_i = \mathbf{E}_i^{\mathrm{T}} \cdot \mathbf{D}_i \cdot \mathbf{E}_i$, $\mathbf{E}_i=(\mathbf{e}_{1,i},\ldots,\mathbf{e}_{n,i})\in\Re^n$ is an orthogonal matrix containing the eigenvectors that define the principal axes of the ellipsoid and $\mathbf{D}_i = diag\left(r_{1,i}^2,\ldots,r_{n,i}^2\right)\in\Re^{n\times n}$ is a diagonal matrix containing the ellipsoid radii. Let s be a sphere with radius r modeling the object O. We can define the *ellipsoidal uncertainty bounding volume of the sphere* s in the i-th projection instant as the set of points $\mathbf{X}_i$ that fulfils the inequality equation (13), where the new matrix $\mathbf{P}_{r,i} = \mathbf{E}_i^{\mathrm{T}} \cdot \mathbf{D}_{r,i} \cdot \mathbf{E}_i$ is obtained from the expressions of the ellipsoid M and the sphere s, being the expression of the diagonal matrix $\mathbf{D}_{r,i}=(\mathbf{D}_i^{1/2}+\boldsymbol{\Lambda}^{-1/2})^2=diag((r_{1,i}+r)^2,\ldots,(r_{\mathrm{n},i}+r)^2)$.

$$(\mathbf{X}_i - \hat{\mathbf{x}}_i)^{\mathrm{T}} \cdot \mathbf{P}_{r,i}^{-1} \cdot (\mathbf{X}_i - \hat{\mathbf{x}}_i) \leq g^2 \tag{13}$$

Consider, for instance, a vehicle modelled with a bounding sphere (Fig. 2(a)) and the uncertainty ellipsoid in its current position (Fig. 2(b)). The ellipsoidal uncertainty bounding volume (13) of the sphere in that instant of time is obtained sliding the ellipsoid around the object boundary (Fig. 2(c)), as done when computing C-obstacles. For complex objects, this is done for every spherical vertex in the s-tope.

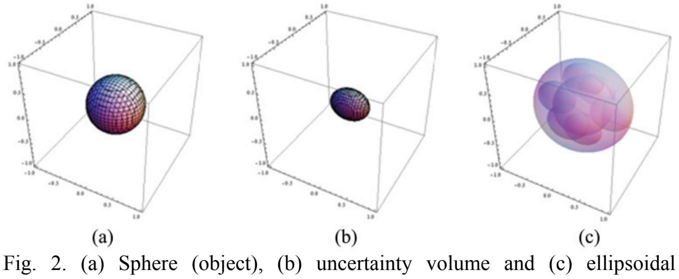

Fig. 2. (a) Sphere (object), (b) uncertainty volume and (c) ellipsoidal uncertainty bounding volume of the sphere.

C. Potential Field Computation

The uncertainty volumes obtained from the trajectory prediction and described by $(\hat{\mathbf{x}}_i, \mathbf{P}_{r,i})$ are considered in the calculation of a new artificial potential field U_i and its derived force $\mathbf{F}_i$. Both are defined in (14) as functions of the vehicle configuration $\mathbf{q}$ for every projection instant i including future instants of time (i>0), therefore adopting a multi-rate scheme. The vehicle configuration $\mathbf{q}$ refers to its position and is usually different from its state $\mathbf{x}$, which can also include velocities (see the vehicle example on section V for a specific case).

$$U_i(\mathbf{q}) = U_{att,i}(\mathbf{q}) + U_{rep,i}(\mathbf{q}) \quad \Rightarrow \quad \mathbf{F}_i(\mathbf{q}) = -\nabla U_i \tag{14}$$

$U_{att,i}$ is an attractive field induced by the goal in i and $U_{rep,i}$ is a repulsive field generated by r obstacles in instant i. Their expressions are given in (15) and (16), where η and ξ are positive scaling factors, ρ_0 represents the limit distance of the repulsive potential field influence, $J \in [0, 1]$ is the *estimation quality index*, $MTD_{i,k}$ is the Minimum Translational Distance (MTD) between the vehicle and the obstacle k at instant i.

$$U_{att,i}(\mathbf{q}) = \tfrac{1}{2} \cdot \xi \cdot \left\| \mathbf{q}_i - \mathbf{q}_{goal} \right\|^2 \qquad U_{rep,i}(\mathbf{q}) = \sum_{k=1}^{r} U_{rep,i,k}(\mathbf{q}) \tag{15}$$

$$\text{with } U_{rep,i,k}(\mathbf{q}) = \begin{cases} \tfrac{1}{2}\eta J^i \left(\dfrac{1}{MTD_{i,k}} - \dfrac{1}{\rho_0} \right)^2 & \text{if } MTD_{i,k} \le \rho_0 \\ 0 & \text{if } MTD_{i,k} > \rho_0 \end{cases} \tag{16}$$

The MTD is defined in [34] as the shortest relative translation of two objects to bring them in contact. It is the separation distance between them if MTD>0 or the penetration distance if MTD<0 (objects colliding). The complexity in the computation of the MTD depends on the model used.

The repulsive potential field generated by an obstacle k, $U_{rep,i,k}$, takes into account the MTD between the ellipsoidal uncertainty bounding volumes of the obstacle and the vehicle obtained from the trajectory prediction, as they are areas where it is likely to find them. Specifically, only ellipsoids from the same projection instant are tested for collisions.

Fig. 3 depicts the *MTD* between ellipsoids obtained from the trajectory prediction at different instants of time within the prediction horizon, for an omnidirectional vehicle and a pedestrian obstacle. In this case, where linear and decoupled kinematic models are used, MTD can be quickly computed by the method in [34], because the uncertainty ellipsoids turn into spheres and the uncertainty volumes are spherical volumes. Hence, the MTD computation is reduced to a Euclidean distance calculus between spheres. If non-linear kinematic models are used, uncertainty volumes are ellipsoids and distance computation between two ellipsoids leads to an algebraic problem of high degree [35] that can eventually be formulated as the problem of finding all roots of a polynomial of degree 24. In this case, the computation of the exact solution would significantly reduce the application of the method in RT and, therefore, a spherical approximation of the uncertainty volume is used for MTD computations.

The potential field generated by (14) propagates in time and remains in the instants without measurements available. This fact greatly improves the planning as there is an anticipation of the future movements of the objects.

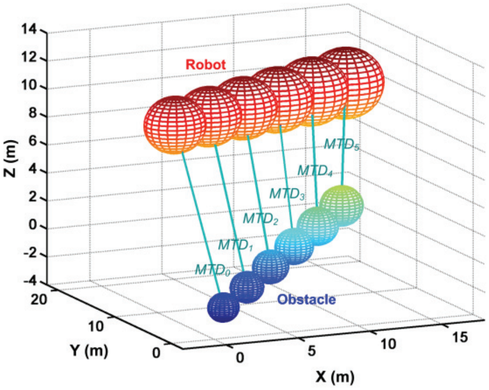

Fig. 3. Uncertainty volumes obtained from the trajectory prediction for an omnidirectional vehicle and a pedestrian obstacle (linear kinematic models).

As a result, at each execution cycle a set of attractive $\mathbf{F}_{att,i}$ and repulsive forces $\mathbf{F}_{rep,i}$ are obtained in the present (i=0) and future (i>0) instants of time. Their expressions are given by (17), where $\boldsymbol{u}_{\boldsymbol{MTDi,k}}$ is a unit vector in the direction of $MTD_{i,k}$ and opposite to the obstacle. Every obstacle k generates a repulsive force in every instant i, whose magnitude decreases as i and $\mathrm{MTD}_{i,k}$ grow. It affects the vehicle motion only from a given security distance ρ_0.

$$\begin{gathered} \mathbf{F}_{att,i}(\mathbf{q}) = -\xi\left(\mathbf{q}_i - \mathbf{q}_{goal}\right) \qquad \mathbf{F}_{rep,i}(\mathbf{q}) = \sum_{k=1}^{r} \mathbf{F}_{rep,i,k}(\mathbf{q}) \\ \text{with } \mathbf{F}_{rep,i,k}(\mathbf{q}) = \begin{cases} \eta J^i \left(\dfrac{1}{MTD_{i,k}} - \dfrac{1}{\rho_0} \right) \dfrac{1}{MTD_{i,k}{}^2} \mathbf{u}_{MTD_{i,k}} & \text{if } MTD_{i,k} \le \rho_0 \\ 0 & \text{if } MTD_{i,k} > \rho_0 \end{cases} \end{gathered} \tag{17}$$

A thorough description of the new potential field and the influence of the parameters can be found in [18].

D. Dealing with Local Minima

The main disadvantage of the PF methods is the presence of local minima for specific obstacle locations that can lead to trap situations [36]. The LE-PFP method maintains this limitation. It may arise when the vehicle moves in the same direction but opposite to an obstacle. However, this problem is only serious when the geometry of the environment is non-convex. In the LE-PFP this problem is not a source of concern since it models obstacles using ellipsoids and the geometry is convex, being the local equilibrium point unstable. Moreover, it can be solved locally with different methods [37, 38].

We use the method in [38] that provides the system with the ability to add repulsive fictitious charges that provoke the elevation of the potential field in local minima positions and compel the vehicle to move away from them.

IV. Integrating Lagrange-Euler & PFP (LE-PFP)

The idea behind trajectory planning using PFP is to include the control actions generated by the conservative and artificial potential field $U_i(\mathbf{q})$ in the trajectory prediction process that takes place in every planning cycle. This way, $U_i(\mathbf{q})$ continuously affects the system and determines its motion as a function of the dynamics properties of the vehicle.

In a configuration space C the general expression of the motion of a system in a configuration $\mathbf{q} \in$C is given by the Lagrange-Euler (LE) equations of motion (18), where the

Lagrangian function L is given by $L(\mathbf{q},\dot{\mathbf{q}})=T(\mathbf{q},\dot{\mathbf{q}})-U(\mathbf{q})$. The non-conservative generalized forces $\mathbf{Q}^{\mathbf{nc}}$ are the friction forces and torques opposing the movement of the vehicle. The first ones are opposite to the linear displacement and the second ones to the angular displacement or rotation.

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\mathbf{q}}}\right)-\frac{\partial L}{\partial \mathbf{q}}=\mathbf{Q}^{\mathbf{nc}} \tag{18}$$

The integration of $U_i(\mathbf{q})$ in the potential term of the LE equations is not straightforward. It requires setting out (18) in terms of the i-th projection instant, as indicated in (19).

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\mathbf{q}}}\right)\bigg|_i-\frac{\partial L}{\partial \mathbf{q}}\bigg|_i=\mathbf{Q}_i^{\mathbf{nc}} \tag{19}$$

Using the Euler approximation for the derivative in the first term of (19) and defining $\partial L/\partial\dot{\mathbf{q}}|_i=\partial L_i/\partial\dot{\mathbf{q}}$, $\partial L/\partial\mathbf{q}|_i=\partial L_i/\partial\mathbf{q}$ the LE formulation in discrete time is:

$$\frac{1}{T}\left[\frac{\partial L_i}{\partial \dot{\mathbf{q}}}-\frac{\partial L_{i-1}}{\partial \dot{\mathbf{q}}}\right]-\frac{\partial L_i}{\partial \mathbf{q}}=\mathbf{Q}_i^{\mathbf{nc}} \tag{20}$$

where L and $\mathbf{Q}^{\mathbf{nc}}$ are considered at the i-th projection instant. The Lagrangian function is given by $L_i(\mathbf{q},\dot{\mathbf{q}})=T_i(\dot{\mathbf{q}})-U_{T\,i}(\mathbf{q})$. The kinetic energy is $T_i(\dot{\mathbf{q}})=\frac{1}{2}\dot{\mathbf{q}}_i^{\mathbf{T}}\mathbf{M}\,\dot{\mathbf{q}}_i$, where the mass matrix $\mathbf{M}$ is constant because the coordinate frame is located at the vehicle center of gravity and moves with it.

It is intended that the vehicle moves under the influence of U_i. Then the total potential energy $U_{T,i}$ is set out including (14) and the potential energy due to gravity $U_{g,i}$:

$$U_{T,i}(\mathbf{q})=U_i(\mathbf{q})+U_{g,i}(\mathbf{q}) \tag{21}$$

If we derive the expression of L_i with respect to the configuration and velocity vectors $\mathbf{q}$, $\dot{\mathbf{q}}_i$ we obtain (22). The conservative generalized forces at i, $\mathbf{Q}_i^{\mathbf{c}}$, refer to the artificial force $\mathbf{F}_i$, to forces derived from $U_{g,i}$ and to torques produced by them all on the vehicle.

$$\frac{\partial L_i}{\partial \mathbf{q}}=\mathbf{Q}_i^{\mathbf{c}} \qquad \frac{\partial L_i}{\partial \dot{\mathbf{q}}}=\mathbf{M}\,\dot{\mathbf{q}}_i \tag{22}$$

Substituting (22) in (20) we obtain (23), where the right term is the generalized force $\mathbf{Q}_i$ and the left term is the approximation of the derivative of $\dot{\mathbf{q}}_i$, i.e., the vector of accelerations $\ddot{\mathbf{q}}_i$. Finally (23) is simplified in (24).

$$\mathbf{M}\frac{[\,\dot{\mathbf{q}}_i-\dot{\mathbf{q}}_{i-1}]}{T}=\mathbf{Q}_i^{\mathbf{c}}+\mathbf{Q}_i^{\mathbf{nc}} \tag{23}$$

$$\ddot{\mathbf{q}}_i=\mathbf{M}^{-1}\mathbf{Q}_i \tag{24}$$

Equation (24) computes the acceleration of the vehicle knowing the forces and torques acting on it as well as its inertial properties in a straightforward manner (without need of numerical integration). It should be provided to the vehicle for it to be able to follow the path determined by U_i.

LE equations produce the desired motion as a sequence of accelerations $\ddot{\mathbf{q}}_i$ (24) during T_h, which have to be converted into the control inputs $\mathbf{u}_i$ required to modify the vehicle motion. This is done by means of kinematic relationships of the type $\mathbf{u}_i$ =$\mathbf{f}$($\ddot{\mathbf{q}}_i$), where $\mathbf{f}$ depends on the particularities of the model (see Fig. 1b and section V for an example).

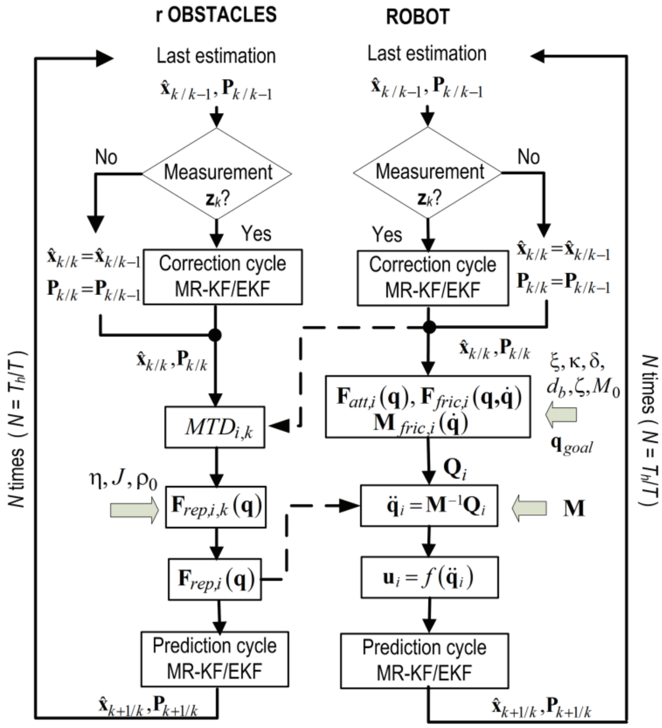
r OBSTACLES
ROBOT
Last estimation
$\hat{\mathbf{x}}_{k/k-1},\mathbf{P}_{k/k-1}$
Last estimation
$\hat{\mathbf{x}}_{k/k-1},\mathbf{P}_{k/k-1}$
No
Measurement $\mathbf{z}_k$?
Yes
Measurement $\mathbf{z}_k$?
No
Yes
$\hat{\mathbf{x}}_{k/k}=\hat{\mathbf{x}}_{k/k-1}$
$\mathbf{P}_{k/k}=\mathbf{P}_{k/k-1}$
Correction cycle MR-KF/EKF
Correction cycle MR-KF/EKF
$\hat{\mathbf{x}}_{k/k}=\hat{\mathbf{x}}_{k/k-1}$
$\mathbf{P}_{k/k}=\mathbf{P}_{k/k-1}$
$\hat{\mathbf{x}}_{k/k},\mathbf{P}_{k/k}$
$\hat{\mathbf{x}}_{k/k},\mathbf{P}_{k/k}$
$\xi,\kappa,\delta,$
d_b,ζ,M_0
$\mathbf{q}_{goal}$
$MTD_{i,k}$
$\mathbf{F}_{att,i}(\mathbf{q}),\mathbf{F}_{fric,i}(\mathbf{q},\dot{\mathbf{q}})$
$\mathbf{M}_{fric,i}(\dot{\mathbf{q}})$
$\mathbf{Q}_i$
η,J,ρ_0
$\mathbf{F}_{rep,i,k}(\mathbf{q})$
$\ddot{\mathbf{q}}_i=\mathbf{M}^{-1}\mathbf{Q}_i$
$\mathbf{M}$
$\mathbf{F}_{rep,i}(\mathbf{q})$
$\mathbf{u}_i=f(\ddot{\mathbf{q}}_i)$
Prediction cycle MR-KF/EKF
Prediction cycle MR-KF/EKF
$\hat{\mathbf{x}}_{k+1/k},\mathbf{P}_{k+1/k}$
$\hat{\mathbf{x}}_{k+1/k},\mathbf{P}_{k+1/k}$
N times ($N=T_h/T$)
N times ($N=T_h/T$)

Fig. 4. Flow chart of the LE-PFP method.

The resulting control inputs are included in the prediction cycle of the appropriate MR-KF/EKF (3) or (4), thus altering the predicted trajectory based on the kinematic model by considering the goal, the obstacles and the system dynamics.

Thereby we obtain the desired sequence of configurations that will act as a reference in the low-level control of the vehicle. They form a local trajectory that meets the constraints of the vehicle dynamics and adapts over time according to the obstacles in the environment.

Notice that the LE equations of motion account for real physical effects on the vehicle that restrict its movement, such as mass, moment of inertia, friction, etc. They are considered in the conservative and no conservative generalized forces. The more realistic these effects are modeled, the better will the vehicle be able to follow the desired trajectory.

A flow chart summarizing the implementation of the LE-PFP method is shown in Fig. 4. Remark that the control input vector $\mathbf{u}_i$ derived from the artificial potential field is considered in the trajectory prediction in a simple fashion.

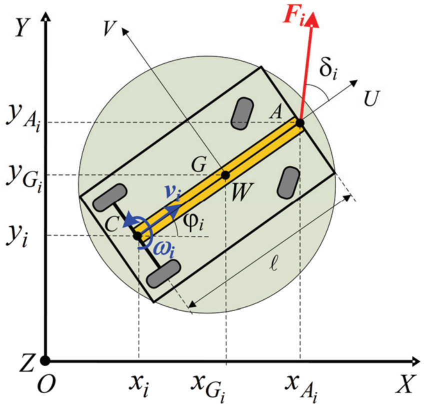

Fig. 5. Dynamic model for a vehicle robot.

V. Example: Vehicle Model

Consider an intelligent vehicle of mass m, with an associated rod dynamic model moving in $\Re^2$ with respect to a fixed frame OXYZ and a local frame GUVW moving with it (Fig. 5). The following simplifying hypotheses have been used for addressing this problem:

1. The vehicle mass m is uniformly distributed along its length, being the transversal dimension negligible with respect to the longitudinal d for dynamic computations.

2. A second order kinematic model of unicycle type (25) is used for trajectory prediction, where the vehicle is represented by point C, located in the middle of the robot chassis. Again, this model is more realistic that the first order model usually employed for robots. This is a stochastic discrete-time system of type (2) where the *jerk* is modeled as a Gaussian and white noise. The state of the vehicle at instant i is given by its position referred to C and its linear and angular velocities, i.e., $\mathbf{x}_i=[x_i\ y_i\ \varphi_i\ \mathrm{v}_i\ \omega_i]^{\mathrm{T}}$. The control vector is composed of linear and angular accelerations referred to C in i, i.e., $\mathbf{u}_i=[a_i\ \alpha_i]^{\mathrm{T}}$.

$$\begin{bmatrix} x \\ y \\ \varphi \\ v \\ \omega \end{bmatrix}_{i+1} = \begin{bmatrix} 1 & 0 & 0 & T\cos\varphi_i & 0 \\ 0 & 1 & 0 & T\sin\varphi_i & 0 \\ 0 & 0 & 1 & 0 & T \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ \varphi \\ v \\ \omega \end{bmatrix}_i + \begin{bmatrix} \frac{T^2}{2}\cos\varphi_i & 0 \\ \frac{T^2}{2}\sin\varphi_i & 0 \\ 0 & \frac{T^2}{2} \\ T & 0 \\ 0 & T \end{bmatrix} \cdot \begin{bmatrix} a \\ \alpha \end{bmatrix}_i + \begin{bmatrix} \frac{T^3}{6}\cos\varphi_i & 0 \\ \frac{T^3}{6}\sin\varphi_i & 0 \\ 0 & \frac{T^3}{6} \\ \frac{T^2}{2} & 0 \\ 0 & \frac{T^2}{2} \end{bmatrix} \cdot \begin{bmatrix} j_a \\ j_\alpha \end{bmatrix}_i \tag{25}$$

$$\begin{bmatrix} x \\ y \end{bmatrix}_i = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ \varphi \\ v \\ \omega \end{bmatrix}_i + \mathbf{v}_i$$

3. For the calculation of $U_{rep,i}$, a circular model with center in G (gravity center) is used enlarged with the uncertainty ellipsoid in i (not shown in Fig. 5), which is approximated by a sphere for fast distance computation with [34].

4. For the sake of simplicity, the forces derived from the potential field U_i are applied at the guidance point A. The geometric relationships between C and A are given by $x_{\mathrm{A}i}=x_i+\ell\cos\varphi_i$, $y_{\mathrm{A}i}=y_i+\ell\sin\varphi_i$ and $\varphi_{\mathrm{A}i}=\varphi_i$.

Note that the general equation (24) refers to the acceleration of G, the vehicle center of gravity. Thus the configuration vector in i, $\mathbf{q}_i=[x_{\mathrm{G}i}\ y_{\mathrm{G}i}\ \varphi_i]^{\mathrm{T}}$, does not coincide with the first three state variables. Besides, the mass matrix is constant $\mathbf{M} = diag(m,m,I_{Gw})$, where $I_{\mathrm{G}w}=(1/12)\cdot m\cdot\ell^2$ is the moment of inertia of the rod with respect to the local axis W.

Let's derive the vehicle equations of motion from (24). The generalized forces $\mathbf{Q}_i$ are both forces $\mathbf{F}(\cdot)$ and moments of forces computed with respect to G, $\mathbf{M}_{\mathrm{G}}(\mathbf{F}(\cdot))$. Thus:

$$\begin{aligned} \mathbf{Q}_i^{\mathbf{c}} &= \mathbf{F}_{att,i}(\mathbf{q}) + \mathbf{F}_{rep,i}(\mathbf{q}) + \mathbf{M}_{\mathrm{G}}\left(\mathbf{F}_{att,i}(\mathbf{q}) + \mathbf{F}_{rep,i}(\mathbf{q})\right) \\ \mathbf{Q}_i^{\mathbf{nc}} &= \mathbf{F}_{fric,i}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{M}_{fric,i}(\dot{\mathbf{q}}) \end{aligned} \tag{26}$$

The non conservative term includes a friction force $\mathbf{F}_{fric,i}$ and a friction moment $\mathbf{M}_{fric,i}$. The friction force is formulated in (27) with the aim of avoiding oscillations near the goal and high velocities. As it is defined in the direction of the linear velocity of the vehicle, this force will not produce any moment with respect to G. Therefore, it does not affect the rotational movement of the robot. The friction moment $\mathbf{M}_{fric,i}$, defined in (28), models friction during rotation, where $\boldsymbol{\omega}_i$ is the angular velocity vector of the vehicle in instant i. $\zeta\in\Re^+$ is a scaling factor and $\mathrm{M}_0\in\Re^+$ is a constant which sets the minimum frictional moment when $v_i=0$ and $\omega_i\neq 0$, i.e, when the robot is rotating around its center of gravity. These friction terms prevent the acceleration to exceed the physical limits of the vehicle and have to be properly tuned in a real application, depending on the vehicle (car, forklift, etc.) and the terrain.

$$\mathbf{F}_{fric,i}(\mathbf{q},\dot{\mathbf{q}}) = -\kappa\cdot\left(1+\delta\cdot e^{-\frac{(\mathbf{q}_i - \mathbf{q}_{goal})^2}{d_b^2}}\right)\cdot\dot{\mathbf{q}}_i \tag{27}$$

$$\mathbf{M}_{fric,i}(\dot{\mathbf{q}}) = -\zeta\cdot(M_0+v_i)\cdot\boldsymbol{\omega}_i \tag{28}$$

Finally, the expression for the generalized acceleration vector $\ddot{\mathbf{q}}_i$ is given by (29), where we call $\mathbf{F}_i = \mathbf{F}_{att,i}+\mathbf{F}_{rep,i}+\mathbf{F}_{fric,i}$. Note that, for simplicity, we have avoided the dependence of $\mathbf{q}$ or $\dot{\mathbf{q}}$ and this convention will be considered from now on.

$$\ddot{\mathbf{q}}_i = \mathbf{M}^{-1}\left(\mathbf{F}_i + \mathbf{M}_{\mathrm{G}}(\mathbf{F}_i) + \mathbf{M}_{fric,i}\right) \tag{29}$$

Breaking down the above expression into components:

$$\begin{bmatrix} \ddot{x}_{\mathrm{G}_i} & \ddot{y}_{\mathrm{G}_i} & \ddot{\varphi}_{\mathrm{G}_i} \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} \frac{F_i\cos(\varphi_i+\delta_i)}{m} & \frac{F_i\sin(\varphi_i+\delta_i)}{m} & \frac{M_{\mathrm{G}}(\mathbf{F}_i)+M_{fric,i}}{I_{\mathrm{G}_w}} \end{bmatrix}^{\mathrm{T}} \tag{30}$$

Since these accelerations are referred to G but the state and control vectors are referred to C (Fig. 5), the following transformation is required:

$$x_i = x_{\mathrm{G}_i} - \tfrac{\ell}{2}\cdot\cos\varphi_i \qquad y_i = y_{\mathrm{G}_i} - \tfrac{\ell}{2}\cdot\sin\varphi_i \qquad \varphi_i = \varphi_{\mathrm{G}_i} \tag{31}$$

Differentiating these equations twice with respect to time we obtain the relations between the accelerations of C and G:

$$\begin{aligned} \ddot{x}_i &= \ddot{x}_{\mathrm{G}_i} + \tfrac{\ell}{2}\cdot\left(\ddot{\varphi}_i\cdot\sin\varphi_i + \dot{\varphi}_i^2\cdot\cos\varphi_i\right) \\ \ddot{y}_i &= \ddot{y}_{\mathrm{G}_i} - \tfrac{\ell}{2}\cdot\left(\ddot{\varphi}_i\cdot\cos\varphi_i - \dot{\varphi}_i^2\cdot\sin\varphi_i\right) \\ \ddot{\varphi}_i &= \ddot{\varphi}_{\mathrm{G}_i} \end{aligned} \tag{32}$$

Now, we can find an explicit expression of the control inputs just substituting these relationships in (33), where the sign of the linear acceleration $sign(a_i)$ depends on the δ_i angle:

$$\mathbf{u}_i = \begin{bmatrix} a_i \\ \alpha_i \end{bmatrix} = \begin{bmatrix} sign(a_i)\cdot\sqrt{\ddot{x}_i^2+\ddot{y}_i^2} \\ \ddot{\varphi}_i \end{bmatrix}^{\mathrm{T}}, \quad sign(a_i)\begin{cases} 1 & \text{if } \delta_i\in\left[-\pi/2, \pi/2\right] \\ -1 & \text{otherwise} \end{cases} \tag{33}$$

Finally, we obtain the expression (34) which provides the control input at instant i as a function of the external forces and moments acting on the vehicle. This vector can be directly used in a MR-KF/EKF to generate a local reference trajectory (positions and velocities) over the prediction horizon in order to head the goal avoiding the obstacles detected by the sensors available in the vehicle.

$$\mathbf{u}_i = \begin{bmatrix} a_i \\ \alpha_i \end{bmatrix} = \begin{bmatrix} sign(a_i)\cdot\sqrt{\frac{F_i^2}{m^2}+\frac{\ell}{m}F_i\left(\omega_i^2\cos\delta_i - \alpha_i\sin\delta_i\right)+\frac{\ell^2}{4}\left(\omega_i^4-\alpha_i^2\cos(2\varphi_i)\right)} \\ \frac{M_{\mathrm{G}}(\mathbf{F}_i)+M_{fric,i}}{I_{\mathrm{G}_w}} \end{bmatrix} \tag{34}$$

With the equations obtained we can calculate the trajectory of any vehicle in a systematic way, if the relations between the vehicle velocity and the velocities of the wheels are known.

VI. Validation Of Results

In the field of path planning there is no standard simulation platform for comparing different reactive planning techniques, although there have been benchmarking initiatives like [39-40] or [41-42] in the field of global motion planning.

With this in mind, the validation of the LE-PFP method has been done in a dynamic environment developed in Matlab© and based on some of the ideas in [39]. It is a 2D four-sided scenario that resembles a pool table, where we can find a configurable number of circular static and mobile obstacles, without collision avoidance of their own. The intelligent vehicle, also circular, travels through the table from an origin to a goal configuration, avoiding the obstacles. The mobile obstacles follow straight paths with constant velocities (from 0.05 to 0.5 m/s) until they reach a table side, at which they change direction simulating a rebound. This environment produces challenging scenarios as the pool table can be cluttered with obstacles moving at high velocities.

The inputs are: obstacles features (number, type, position, dimension, speed, motion direction, kinematic model); vehicle models (kinematic and dynamic); control period T, execution period of the LE-PFP algorithm T_d and prediction horizon T_h; origin and goal configurations. Distances to obstacles are provided contaminated with Gaussian white noise.

The outputs are: the vehicle resulting trajectory; number of collisions, if any; velocity and acceleration profiles; execution time of the planning algorithm.

The LE-PFP is compared against the PF method [3] using a depth-first planning strategy, where the total force indicates the most promising motion direction and the vehicle moves forward into segments of equal length in each period. It is assumed that a tracking module provides obstacle information and linear kinematic models given by (35) are associated to them, simulating pedestrians or omnidirectional vehicles.

$$\begin{bmatrix}x\\y\\v_x\\v_y\end{bmatrix}_{i+1}=\begin{bmatrix}1&0&T&0\\0&1&0&T\\0&0&1&0\\0&0&0&1\end{bmatrix}\cdot\begin{bmatrix}x\\y\\v_x\\v_y\end{bmatrix}_i+\begin{bmatrix}\frac{T^2}{2}&0\\0&\frac{T^2}{2}\\T&0\\0&T\end{bmatrix}\cdot\boldsymbol{w}_i \qquad \begin{bmatrix}x\\y\end{bmatrix}_i=\begin{bmatrix}1&0&0&0\\0&1&0&0\end{bmatrix}\cdot\begin{bmatrix}x\\y\\v_x\\v_y\end{bmatrix}_i+\mathbf{v}_i \qquad (35)$$

A real navigation application requires data acquisition and sensor fusion as well as auto-localization and map building, operations that introduce accumulative delays that result in a late latency of the planning method that may end in collision. For that reason, a situation with a 3s lack-of-measurements has been simulated assuming a laser rangefinder and a post-processing stage (range 1-3m, 180º, 10 Hz, 5cm accuracy).

Fig.6 represents three simulation instants for the PF method [3] implemented as explained previously. The vehicle (blue circle) should reach the goal (red point) without colliding with any of the 3 mobile obstacles (pink circles). Fig. 6(a) shows the vehicle and the obstacle 1 dangerously close. The vehicle has not yet begun the avoidance, since there were no measurements available in the last 3s. In Fig. 6(b, c) the vehicle performs an avoidance maneuver but it is certainly too late since it ends up colliding with obstacle 1.

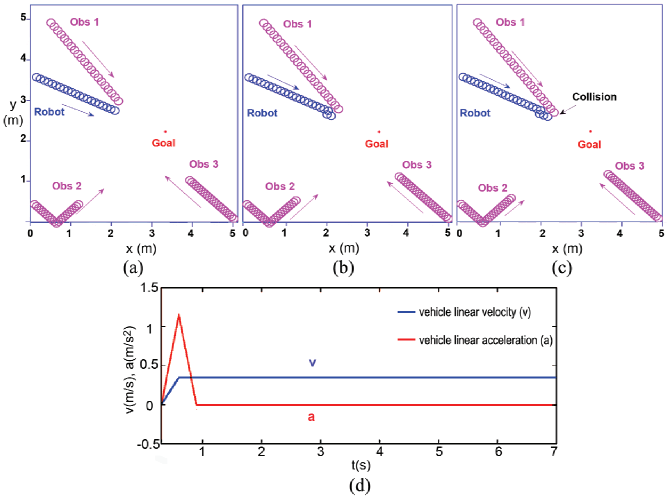

Fig. 6. PF simulation (T=0.3s, T_d=3s): (a) vehicle approaching obs. 1, (b) (c) vehicle beginning the avoidance & ending in a collision, (d) kinematic profile.

The situation has been replicated using the LE-PFP method (T=0.3s, T_d=3s, T_h=4s) and the model explained in Section V depicted (Fig. 7 (a,b)) as well as the trajectory generated to reach the goal (Fig. 7(c)) and the profiles for linear and angular velocities and accelerations (Fig. 7(d)). Uncertainties in vehicle and obstacles positions are not depicted for clarity. In Fig. 7(a) the vehicle and the obstacle 1 are close but do not collide, as the vehicle has made a turn. It simultaneously performs a deceleration and a sudden turn to avoid obstacles 2 & 3, with success (Fig. 7(b)). Fig. 7(c) represents the path followed by the vehicle, which leads it to the goal without colliding with any of the three obstacles (see enclosed video "video1_Mtlb.mov" at http://ieeexplore.ieee.org).

Note that LE-PFP uses the kinematic and dynamic models of the vehicle to compute the reference velocities for the control. This is evident in the representation of the velocity (Fig. 7(d)) and the trajectory (Fig.7(c)). The vehicle starts with zero velocity at position (0, 3.7) and soon reaches a nearly constant velocity because it is far from any obstacle. From about 12s it begins to slow down and turn anticipating the

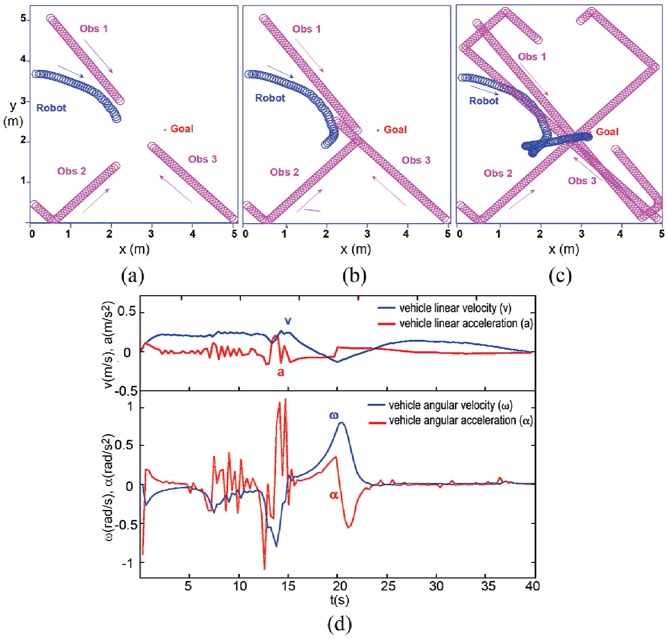

Fig. 7. LE-PFP simulation (T=0.3s, T_d=3s, T_h=4s, $\mathbf{R}$=$diag$(0.007)m^2): (a) obs. 1 avoidance, (b) obs. 2&3 avoidance, (c) trajectory, (d) kinematic profile.

avoidance of obstacle 1. Next, the vehicle accelerates again to overcome obstacle 3 (at 14s). Around the 20s the vehicle makes an effort in order to turn toward the goal and finally breaks reaching zero velocity at the end of the simulation, stopping at the goal. This behavior is not seen in the PF method (Fig. 6(d)), which always provides a constant velocity reference, as the obstacle is too fast to be detected on time.

To demonstrate the performance of the LE-PFP in cluttered and challenging environments, a new scenario with 20 obstacles is included (see video “video2_Mtlb.mov”), where the vehicle has to reach two sequential goals and obstacles perform sudden changes, as pedestrians or omnidirectional robots would do. This kind of uncertain behavior is considered in the noise terms of the kinematic models. In those instants, although the MR-KF prediction is poor, the filter is able to reduce the predicted trajectory uncertainty after a few new measurements are taken. The setting and kinematic profiles of the resulting trajectory are depicted in Fig. 8

Remark that parameters ξ, η, δ, κ, d_b, ζ, ρ_0 have to be tuned to fit the features of the system (see [18] for further information). In very uncertain environments due to non accurate measurements and high estimation errors the vehicle may stop, if there is no clear path to follow, or even collide, if sensors are not fast or accurate enough.

VII. High Speed Simulation Application

In this section, a high speed simulation scenario, presented in [43], has been used for the LE-PFP testing. It is part of a simulation environment for testing planning algorithms, very useful as a previous step in the implementation of navigation algorithms on real platforms and already used in [44, 45].

The modeling and programming of the simulation environment has been developed entirely in C++, using the Dark GDK library [46]. The LE-PFP method has been implemented using the Newmat mathematical library [47], which facilitates its subsequent integration into different experimental platforms (Windows-based, Linux-based…). The simulation code has been designed to include algorithms as independent functions within a library. Thus, the LE-PFP method has been implemented as an independent module.

The scenario is a four-lane highway where vehicles travel both ways (Fig. 9). A motorbike represents the intelligent vehicle and different car models simulate the obstacles. Linear kinematic models (35) are used for the obstacles and a vehicle model (25) is used for the intelligent vehicle. The method uses spherical models for the objects, along with their uncertainties.

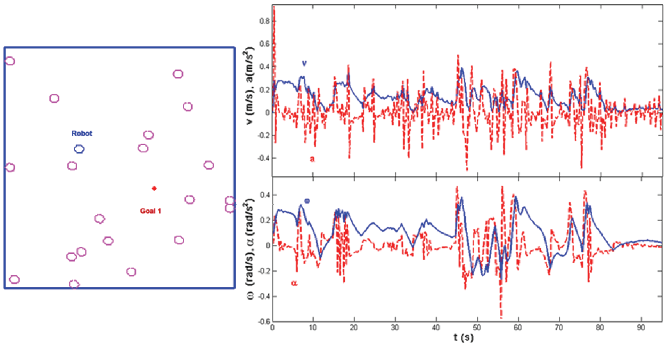

Fig. 8. LE-PFP simulation (T=0.3s, T_d=3s, T_h=4s, **R** ranging from diag($3\cdot10^{-4}$) m^2 to diag(0.01) m^2) with 20 obstacles modeled by linear models. Setting (left) and kinematic profiles (right).

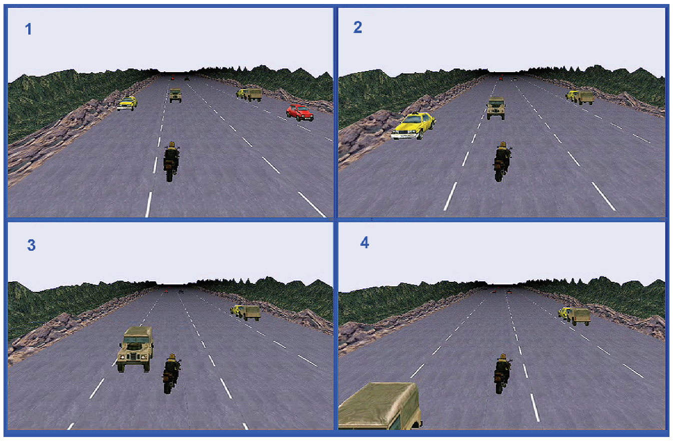

Fig. 9. High-speed simulation environment showing a motorcycle avoiding a car in different snapshots.

The case of a vehicle following a predefined straight-line trajectory is shown. When a car occludes its trajectory or is approaching from the rear side, the LE-PFP method computes a new trajectory to avoid it. This trajectory fully meets the kino-dynamic requirements of the vehicle.

This is a challenging scenario where safety is a key concern. As the environment represents an Intelligent Highway System, the generated trajectory is always a lane changing maneuver, being the goal located at a constant distance ahead from the vehicle. The different periods of the algorithm are T_d=2s, T_h=3s and T=0.1s. **R** ranges from $diag(3\cdot10^{-4})$m^2 to $diag(0.01)$m^2 and the values of parameters related to the potential field and friction forces are $\xi=10^3$, $\eta= 8\cdot10^7$, δ=100, κ=200, d_b=300, ζ=500, ρ_0=400, J=0,95. Finally, the mass of the robot is m=1000 kg and the length considered is ℓ=4 m.

An example of avoidance is displayed in the sequence of images in Fig. 9 and in the video “video3_Highway.mov”.

The method achieves very good performance and execution time in Matlab and C++ applications. The execution time is O(n) with n the number of obstacles. The mean time for detected obstacle is around 9 ms in a Pentium IV, 3 GHz, 1GB RAM. Hence, in a scenario with 100 obstacles the algorithm would obtain a local trajectory in less than 1s, which is very suitable for RT applications.

VIII. Conclusions And Future Work

This paper presents a new formulation of a predictive and multi-rate reactive planning method for dynamic and uncertain environments. It is based on potential fields and has some advantages when compared to other methods:

- Path planning and trajectory planning are combined in a single method with an integral formulation.
- It incorporates the uncertainty in measurements and the predicted trajectories of the obstacles and the vehicle in the potential field computation.
- Its formulation provides a holistic framework: it can be used for any kind of vehicle, governed by any kinematic and dynamic model. Kinematics is considered in the prediction process and dynamics is introduced via the LE formulation.

• Trajectories for holonomic and non-holonomic vehicles are generated using the same formulation.

As a result, a local trajectory is naturally obtained that fully meets the kinodynamic vehicle constraints and adapts over time according to the obstacles in the environment.

Future extensions of this work include the development of a systematic methodology to tune the terms in the method; the introduction of rules; the simulation of mixed traffic scenarios with complex situations (like head-on scenarios) and different obstacle models, cooperative planning, real implementations, other estimation filters (like particle filters), and the application to mobile manipulators or vehicles with trailers.

REFERENCES

[1] Latombe, J. C. 1991. Robot motion planning. Kluwer Academic Pub.

[2] J. Minguez, F. Lamiraux, J-P Laumond. "Motion Planning and Obstacle Avoidance", *Handbook of Robotics*, Springer, pp. 827-852, 2008.

[3] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The Int. Journal of Robotics Research*, 5, 1, pp. 90-98, 1986.

[4] I. Ulrich and J. Borenstein, "VFH*: Local obstacle avoidance with look-ahead verification," in Proc. *IEEE ICRA*, 2000, pp. 2505-2511.

[5] J. Minguez and L. Montano, "Nearness Diagram Navigation (ND): Collision Avoidance in Troublesome Scenarios," *IEEE Transactions on Robotics and Automation,* 20, 1, pp. 45-59, 2004.

[6] R. Simmons, "The Curvature-Velocity Method for local obstacle avoidance," in *Proc. IEEE ICRA*, 1996, pp. 3375-3382.

[7] D. Fox, W. Burgard and S. Thrun, "The Dynamic Window approach to collision avoidance". *IEEE Robotics&Automation*, 4(1), pp.23-33, 1997.

[8] P. Ögren and N.E .J. Leonard, "A convergent dynamic window approach to obstacle avoidance," *IEEE Trans. Robotics*, 21, 2, pp. 188-195, 2005.

[9] A. Asensio and L. Montano, "A kinematic and dynamic model-based motion controller for mobile robots," *Proc. 15th IFAC World Conf.,*2002.

[10] J.C. Alvarez, A. Shkel and V. Lumelsky, "Accounting for Mobile Robot Dynamics in Sensor-Based Motion Planning: Experimental Results," in *Proc. IEEE ICRA*, pp. 2205-2210, 1998.

[11] A. A. Masoud, "Dynamic trajectory generation for spatially constrained mechanical systemas using harmonic potential fields," in *Proc. IEEE ICRA*, vol. 3, pp. 2944 – 2951, 2007.

[12] Q. Cao, Y. Huang and J. Zhou, "An evolutionary artificial potential field algorithm for dynamic path planning of mobile robot," in *Proc. IEEE/RSJ IROS*, pp.3331–3336, 2006.

[13] Z. Shiller, F. Large, S. Sekhavat, "Motion planning in dynamic environments: obstacles moving along arbitrary trajectories," in *Proc. IEEE ICRA*, vol.4, pp. 3716 -3721, 2001.

[14] R. Philippsen, S. Kolski, K. Macek and B. Jensen."Mobile Robot Planning in Dynamic Environments and on Growable Costmaps," in *Proc. IEEE ICRA*, 2008.

[15] S-Y. Chung and H-P. Huang, Robot Motion Planning in Dynamic Uncertain Environments, *Advanced Robotics*, **25**, 849–870, 2011.

[16] Masoud, A.A. "A harmonic potential field approach with a probabilistic space descriptor for planning in non-divisible environments". *Proc. IEEE ICRA*, vol. 3, pp. 2944 – 2951, 2009.

[17] M.C. Mora, R. Pizá and J. Tornero, "Multirate obstacle tracking & path planning for intelligent vehicles," in *Proc. IEEE IV*, pp. 172-177, 2007.

[18] M.C. Mora and J. Tornero, "Path planning and trajectory generation using multi-rate predictive artificial potential fields," in *Proc. IEEE/RSJ IROS,* pp. 2990-2995, 2008.

[19] J. Tornero, J. Salt and P. Albertos, "LQ optimal control for multirate sampled data systems," Proc.*14th IFAC World Conf.*, pp.211-216, 1999.

[20] R. Pizá , "Modelado del entorno y localización de robots móviles autónomos mediante técnicas de muestreo no convencional," phD Thesis, Universidad Politécnica de Valencia, Valencia, 2003.

[21] D. Helbing and P. Molnar. Social force model for pedestrian dynamics. Physical Review E, 51, pp. 4282-4286, 1995.

[22] S.P Hoogendoorn and W. Daamen. Microscopic parameter identification of walker models and its implications to pedestrian flow modeling. In T RR: J. Transportation Research Board, 2124, pp. 1-13, 2006.

[23] T. Lakoba, D. Kaup, N. Finkelstein. Modications of the Helbing-Molnn'ar- Farkas-Vicsek Social force model for pedestrian evolution. Simulation, 81, pp.339-352, 2005.

[24] S. Pellegrini , A. Ess , K. Schindler, L. vanGool, "You'll never walk alone: modeling social behavior for multi-target tracking",in Proc. Int. Conf. on Computer Vision, pp.261–268, 2009.

[25] M. Luber, J. Stork, G. Tipaldi, K. Arras, "People tracking with human motion prediction from social forces", Proc. IEEE ICRA,464–469, 2010.

[26] A. Jazayeri, H. Cai, J.Y. Zheng, and M. Tuceryan, "Vehicle Detection and Tracking in Car Video Based on Motion Model", IEEE Transactions on Intelligent Transportation Systems 12(2), 2011.

[27] J. Borenstein, H.R.Everett, L.Feng, D.Wehe, "Mobile robot positioning: sensors and techniques*," J. Robotic Systems* 14(4), pp. 231-249, 1997.

[28] H.I. Christensen, G.D. Hager, "Sensing and Estimation", *Handbook of Robotics*, Springer, pp. 87-107, 2008.

[29] Y. Bar-Shalom, X. Rong Li and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*, John Wiley & Sons, Inc., 2001.

[30] L.R. Paradowski, "Uncertainty ellipses and their application to interval estimation of emitter position,". *IEEE Transactions on Aerospace and Electronic Systems*, 33(1), pp. 126-133, 1997.

[31] R. Deutsch, *Estimation Theory*. Prentice-Hall, 1965

[32] G.J. Hamlin, R.B. Kelley, and J. Tornero, "Efficient distance calculation using the spherically-extended polytope (s-tope) model," in *Proc. IEEE ICRA*, vol. 3, pp. 2502–2507, 1992.

[33] E.J. Bernabeu and J. Tornero, "Optimal Geometric Modeler for Robot Motion Planning". *J.of Robotic Systems*, 17(11), 593-608, 2000.

[34] E.J. Bernabeu and J. Tornero, "Hough transform for distance computation and collision avoidance". IEEE Transactions on Robotics and Automation, 18(3), pp. 393-398, 2002.

[35] C. Lennerz and E. Schömer, " Efficient distance computation for quadratic curves and surfaces, " *Proc. Geometric Modeling and Processing*, IEEE Press, pp. 60–69, 2002.

[36] Y. Koren and J. Borenstein, "Potential field methods and their inherent limitations for mobile robot navigation," in *Proc. IEEE ICRA*, pp. 1398-1404, 1991.

[37] H. Haddad, M. Khatib, S. Lacrox, and R. Chatila, "Reactive navigation in outdoor environments using potential fields," in *Proc. IEEE ICRA,* pp. 1232-1237, 1998.

[38] D. Maravall, J. de Lope and F. Serradilla, "Combination of model-based and reactive methods in autonomous navigation," in *Proc. IEEE ICRA,* vol. 3, pp. 2328 – 2333, 2000.

[39] Rañó and J. Mínguez, "Steps towards the automatic evaluation of robot obstacle avoidance algorithms," in *Proc. IEEE/RSJ IROS*, 2006.

[40] J.L. Jiménez, I. Rañó and J. Mínguez, "Advances in a framework for automatic evaluation of obstacle avoidance algorithms," in *Proc. IEEE/RSJ IROS*, 2007.

[41] Project *Motion Planning Puzzles*, Parasol Lab. 2009. Available : http://parasol-www.cs.tamu.edu/groups/amatogroup/benchmarks/mp.

[42] European Robotic Research Network: Benchmarking Initiative, 2009. Available : http://www.euron.org/activities/benchmarks/index.html.

[43] M.C. Mora and J. Tornero, "Non-structured simulation environment for collision detection and obstacle avoidance algorithm testing", in *Proc. ISC*, pp. 223-227, 2003.

[44] M.C. Mora and J. Tornero, "Evitación de obstáculos con realimentación de fuerza en un sistema de simulación y teleoperación de robots móviles", in *Proc. V Workshop en Agentes Fisicos*, pp. 89-99, 2004

[45] N. Montés, M.C. Mora and J. Tornero, "Trajectory generation based on rational bezier curves as clothoids," in *Proc. IEEE IV*, pp. 505-510, 2007

[46] Dark Game Development Kit. 2009. http://gdk.thegamecreators.com/

[47] R. Davies, Newmat matrix library. Available : http://www.robertnz.net/