Dynamic Potential Field Generation Using Movement Prediction

Conference Paper · July 2012

DOI: 10.1142/9789814415958_0098

5 authors, including:

Raul Acuña
Technische Universität Darmstadt

Ángel Terrones
Simon Bolívar University

Novel Certad
Simon Bolívar University

Gerardo Fernandez
Simon Bolívar University

CLAWAR 2012 – Proceedings of the Fifteenth International Conference on
Climbing and Walking Robots and the Support Technologies for Mobile Machines,
Baltimore, MD, USA, 23 – 26 July 2012

DYNAMIC POTENTIAL FIELD GENERATION USING MOVEMENT PREDICTION

R. ACUÑA, A. TERRONES, N. CERTAD-H, L. FERMÍN-LEÓN and
G. FERNÁNDEZ-LÓPEZ

Simón Bolívar University, Mechatronics Group, ELE-328, Sartenejas 1080-A
Miranda, Venezuela
**E-mail: racuna@usb.ve*

Path-planning in dynamic environments while meeting safety requirements for robots and humans is an open problem in robotics. The successful navigation in this kind of environments requires a certain level of anticipation to the future behavior of moving objects. In this paper we propose the use of movement prediction in the generation of dynamic artificial potential fields (DAPF), which will allow the robot to navigate in highly dynamic environments with a major degree of safety and effectiveness, especially in cases where the obstacles and objectives move at higher velocities than the controlled robots. Our approach is based on previous works on potential and velocity fields with additional considerations to avoid the well known local minimum problem and to anticipate the predicted path of objects, our solution tries to maintain the reactive characteristics of the artificial potential fields. The proposed method was tested in a holonomic simulation with 100% better results for the same scenarios than the original fields without prediction.

Keywords: Artificial Potential Field; Path-Planning; Local Minimum Problem; Velocity Field Generation; Movement Prediction; Autonomous Navigation

1. Introduction

The navigation using Artificial Potential Fields (APF) was presented for the first time by Khatib with the introduction of FIRAS, this function was designed to achieve real-time collision avoidance in unknown environments for manipulators and mobile robots.[1] This original idea has been in continuous development and adapted through the years in the field of robotics. However, there were many problems inherent to the APF approach which were summarized by Koren *et al.*[2] In an attempt to solve those problems, new methods emerged to modify the fields in order to cope with their limitations. The *Virtual Force Field Method* (VFF), the *Vector Field Histogram* (VFH)[3] and the enhanced VFH*[4] for local obstacle avoidance are some examples.

Despite its limitations, the APF are still widely used due to its real-time performance in many applications like planetary exploration rovers,[5,6] or in foraging tasks.[7] In other related work Li and Horowitz proposed the *Passive Velocity Field Control* (PVFC), in this method the motion task for manipulator control is encoded as a velocity field,[8] this method was improved in consecutive works by Li and Moreno.[8,9] However, this solution was inefficient in environments that change rapidly. Therefore new ways of dealing with the generation of references in an active fashion emerged, primary, in the works of Bruijnen,[10] ,Estévez[11] and more recently Pérez-D'Arpino,[12] but it is still considered as an open problem due to the limitless variations in dynamic settings. One of the many problems in dynamic environments is the likely presence of moving objects with greater or comparable velocities than the top speed of the robot. This creates a safety hazard, since in some cases the APF planner may take the robot to an area believed safe but that will be occupied by a fast moving object in the near future, in this scenario the robot may not be able to escape that area in the next iterations and a collision will occur. In this work a prediction scheme is introduced which modifies the fields in order to anticipate the future positions of moving objects on the workspace. This allows the controlled robots a mechanism to avoid the regions that will be occupied by moving obstacles, also this method is used to predict the position of goals and modify the fields to anticipate them.

2. Methodology

The solution proposed in this paper aims to solve general problems of navigation with motion prediction. In order to illustrate the most typical scenarios two main cases were selected: Predictive obstacle avoidance with and without defined objectives. In the predictive obstacle avoidance scenario without objetives, one robot lies on the path of a fast moving obstacle and it needs to avoid a collision. On the second case, the robot is traveling towards a defined objetive in a course that will pass through the path of a fast moving obstacle, once again the robot has to avoid the collision and also get to the goal. The proposed method is divided into two main parts: The predictor, responsible of the indexing and prediction of moving obstacles and goals; and the planner which defines the attractive and repulsive fields according to the position of the robot, the goals, the obstacles and their predicted paths. This method assumes the presence of a subsystem which provides the location information of all the objects in the workspace in each iteration. The primary experiments were implemented using a Python^{TM}

simulation with an holonomic platform. Additional tests were implemented in the robot development environment Webots^{TM} with physics using a simulated differential wheels robot Amigobot. The maximum speed and other physical properties of the simulated robot in both cases are similar to the those of the real Amigobot.

The proposed solution is organized as in Fig. 1.

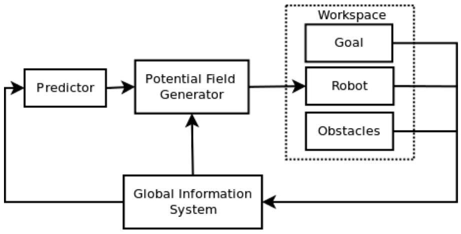

Fig. 1: Block Diagram of the predictor and potential field generation system.

3. Predictor and Potential Fields Definition

The system faces two constrains,[13] the response time available to plan a safe movement (which is a function of the system dynamics) and a temporal restriction about the certainty of the prediction in the future. In other words, it is imperative to plan rapidly but you can not plan far away in the future. In order to cope with this effect it is possible the use of a precise short term predictor, an inaccurate long term predictor or hybrid solutions. In any case, previous data is needed in order to predict whether a system model or historical data information. In our solution we assume that we do not know anything about the moving objects, therefore we can not make a prediction without studying their past actions. In this scenario the *Recursive Motion Function* (RMF)[14] is an ideal candidate, because it creates a model based on the historical positions of the indexed objects, this motion function can express a large number of simple and complex movement types like polynomials, ellipses, circles, sinusoids, etc. In the RMF algorithm, any polynomic function of degree D can be converted to a lineal recurrence after $D + 1$ differentiations. Based on this information Tao proposed the following recursive function:[14] $o(t) = \mathbf{C}_1 \cdot o(t-1) + \mathbf{C}_2 \cdot o(t-2) + \cdots + \mathbf{C}_f \cdot o(t-f)$. Wich can be converted to matrix form, defined as the motion state $\mathbf{S}_o(t) = \mathbf{K}_o \cdot \mathbf{S}_o(t-1)$

Where $\mathbf{C}_i(1 \leq i \leq)$ is a $d \times d$ constant matrix (d is the dimension of the space), and f is an important parameter called the retrospect. The *motion* *state* $\mathbf{S}_o(t)$ is defined for an object o at time t as a vector representing its location at the f most recent timestamps. Where $\mathbf{K}_o$ is a constant $(d \cdot f) \times (d \cdot f)$ motion matrix for o.

In our solution we are using $d = 2$, and using the two most recent locations (retrospect $f = 2$) then equation 3 becomes:

$$\begin{bmatrix} o(t) \cdot \mathbf{x}_1 \\ o(t) \cdot \mathbf{x}_2 \\ o(t-1) \cdot \mathbf{x}_1 \\ o(t-1) \cdot \mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{k}_{11} & \mathbf{k}_{12} & \mathbf{k}_{13} & \mathbf{k}_{14} \\ \mathbf{k}_{21} & \mathbf{k}_{22} & \mathbf{k}_{23} & \mathbf{k}_{24} \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} o(t-1) \cdot \mathbf{x}_1 \\ o(t-1) \cdot \mathbf{x}_2 \\ o(t-2) \cdot \mathbf{x}_1 \\ o(t-2) \cdot \mathbf{x}_2 \end{bmatrix} \tag{1}$$

The problem is finding the matrix K. We use the method of motion estimation proposed by Tao[14] using the actual location $\mathbf{l}_o(t)$ of the object o at time t and the h most recent timestamps to solve $S \cdot \mathbf{k}_l(t) = l$. In order to find the solution for this equation we need to find the inverse of S, Tao proposed *singular value decoposition* (SVD) but in our work we also tried other faster iterative methods like the gradient conjugate suitable for large retrospects f.

In our holonomic simulation we tested different types of movements like linear, polynomial and circular. In those cases a retrospect of $f = 5$ represented accurately the motion as indicated by Tao. However, when we add some noise to the input data the results change dramatically. In Fig. 2 the motion function of a sinusoid is predicted using $f = 5$ and the prediction is almost perfect. In Fig. 3 we add noise to the same sinusoid and in this case the predictor fails to estimate the exact motion behavior and the prediction is less accurate. If computational time is not an issue one way to solve this problem is to increase the value of the retrospect f, this mitigates the effect of the noise, in Fig. 4 we use for the same noisy data a retrospect $f = 40$, this prediction is less smooth but is more accurate than the case with a small retrospect.

Also, in our Webots simulation we implemented a Kalman filter that greatly reduce the effect of the noise in the prediction. In terms of computational time, an increment in the retrospect produces a larger S and the SVD computation time increases. For a retrospect $f = 100$ our predictor implementation in a Core 2 Duo CPU took $8ms$ while a large retrospect $f = 100$ took $30ms$. For single object tracking and prediction this may not be a problem, but with several objects this could be excessively high. An alternative is to solve the inverse of S using an iterative method like conjugate gradient and stop at n iterations, this produces an estimate of the inverse but the computational time is decreased with similar results,

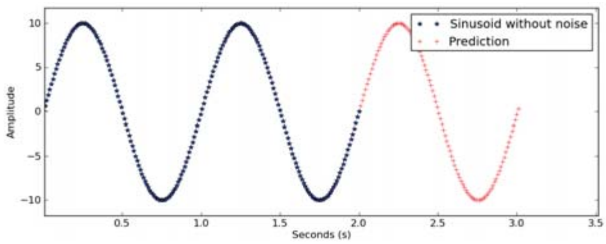

Fig. 2: Sinusoid prediction. Predicion Horizont $Ph = 100$, $f = 5$ and $h = 100$.

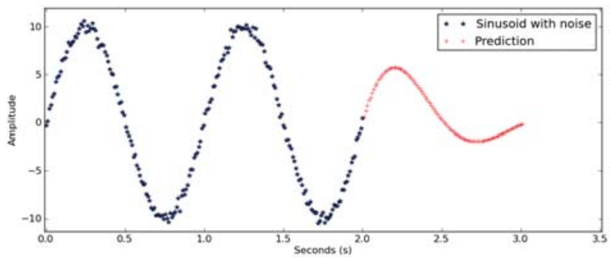

Fig. 3: Sinusoid with noise prediction. Predicion Horizont $Ph = 100$, $f = 5$ and $h = 100$.

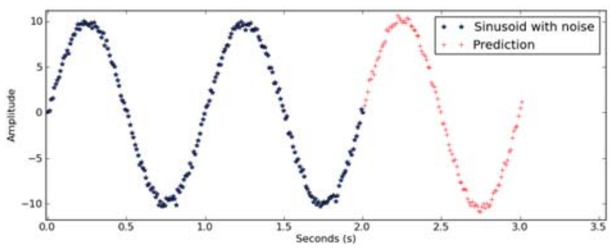

Fig. 4: Sinusoid with noise prediction. Predicion Horizont $Ph = 100$, $f = 40$ and $h = 100$.

for a retrospect of $f = 100$ the time to calculate the conjugate gradient was $8ms$.

The generation of vector references for the robot is based in potential surfaces. Basic potential functions for obstacles, goals and walls are described in the work of Estevez.[11] For the obstacles we use Gaussian Hills, which have limited range of action and continuous gradient. For the goals Conical Attractors are used that provide constant radial movement. The walls are considered obstacles and they have the same type of potential surface, with a maximum in the position of the wall and a Gaussian decay in a direction perpendicular to the wall. In our method the predicted path of the obstacles creates a repulsive potential field. Given a prediction $P(t)$ of dimension $d \times Ph$, for an object o, where d is the space dimension and Ph is the prediction horizon. A Gaussian potential field with σ_x and σ_y according to the size of the object o is created for each $P(t)$ $\forall t \in [t, tPh]$. The individual contribution of each Gaussian Surface in each $P(t)$ is summed with the consecutive surfaces, this creates a Gaussian potential path along the predicted route of the object o. Also, a similar approach is used to avoid the local minima problem, a Gaussian path along recent historical positions of the robot favors the continuous movement of the robot towards the goal.

4. Results and Discussion

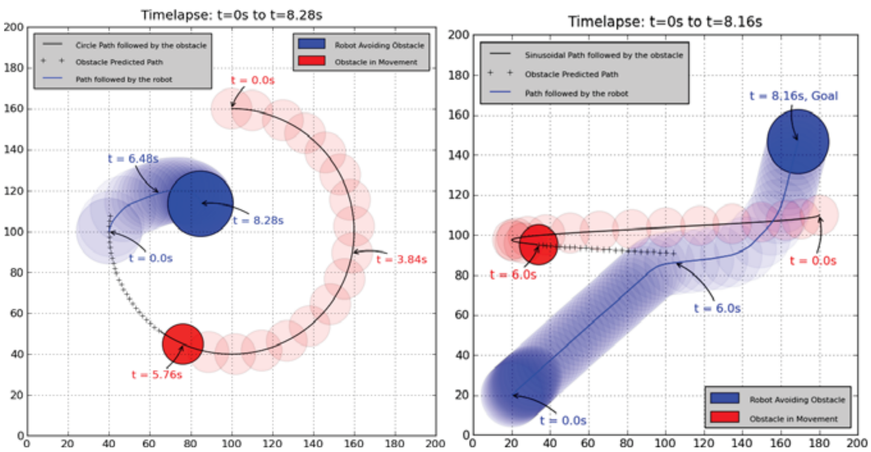

Fig. 5: Predictive obstacle avoidance. Left: without goals, Right: with goal at (170,150).

In Fig. 5 a fast moving obstacle (red) travels in a collision course to the slower Robot (blue), in Fig. 6 one obstacle travels in the path of the robot and another travels perpendicular. The predicted position of the obstacles and the generated fields allow the robot to prevent the collision, when this same tests where performed without prediction a collision always occurred. However the success of the proposed method depends on several factors, the training horizon H, the retrospect f, the prediction Horizon Ph and finally

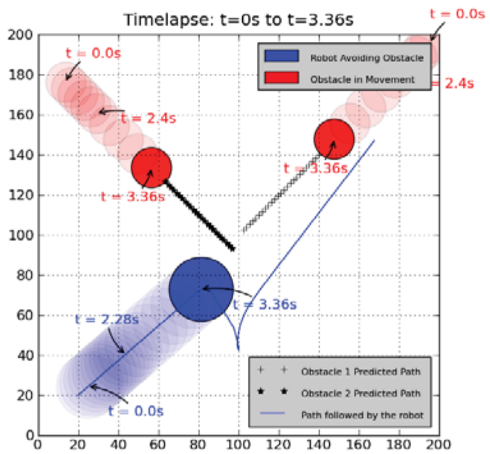

Fig. 6: Predictive obstacle avoidance. With Goal and Two Obstacles.

the weight and size of the potential path. The optimum Ph in this method is an open issue since it depends on both the relative speed between the robot and the obstacle and the relative direction of movement, which is not easy to determine since the movements are not always linear. A large value in Ph may increase the available time to successfully avoid the obstacle but a long predicted path will affect a large area of the workspace that may not be relevant and also increase the uncertainty of the prediction in the future. From our tests we conclude that it is safe to set Ph equal to the value of H. The size and weight of the Gaussian Potential Path PGP may change the outcome even if the prediction is accurate. If the Gaussian area and weight is too small the robot will not avoid the obstacle in time, in contrast a large area of effect will repel any robot in a vast area that in reality may be safe. A good value for the size of the PGP is the same width as the maximum of the obstacle. Finally if one obstacle suddenly changes direction, the prediction using RMF will fail and will be very unstable until H samples of the new type of movement have been indexed. In the meantime it is necessary to detect the moment of a sudden change in the movement pattern and disable the predictor.

Previous techniques for navigation using potential fields usually assume static obstacles or slow moving obstacles. Our paper presents a reactive solution using Gaussian repulsion zones based on the predicted positions of the obstacles to ensure collision-free trajectories in scenarios with fast moving obstacles. We believe that this method provides a safer reactive planner than other methods and open many doors for future research on the effect of the variations of the parameters to ensure safety and goal achievement.

References

1. O. Khatib, Real-time obstacle avoidance for manipulators and mobile robots, *The international journal of robotics research* **5**, p. 90 (1986).
2. Y. Koren, S. Member, J. Borenstein and A. Arbor, Potential Field Methods and Their Inherent Limitations for Mobile Robot Navigation, *Proceedings of the IEEE* , 1398 (1991).
3. J. Borenstein and Y. Koren, Real-time Obstacle Avoidance for Fast Mobile Robots in Cluttered Environments 1, *Robotics* , 572 (1990).
4. I. Ulrich and J. Borenstein, VFH*: Local Obstacle Avoidance with Look-Ahead Verification, *Robotics* , 2505 (2000).
5. M. Massari and G. Giardini, Autonomous navigation system for planetary exploration rover based on artificial potential fields, *Control of Systems* (2004).
6. J. Hagelbäck and S. J. Johansson, Using Multi-agent Potential Fields in Real-time Strategy Games, *Strategy* , 631 (2008).
7. O. Simonin, Construction of Numerical Potential Fields with Reactive Agents, *Science* , 1 (2005).
8. P. Y. Li and R. Horowitz, Passive velocity field control of mechanical manipulators, *IEEE Transactions on Robotics and Automation* **15**, 751 (1999).
9. J. Moreno and R. Kelly, On manipulator control via velocity fields, *Proceedings of the 15th IFAC World Congress*, **1**, 4 (2002).
10. D. Bruijnen, J. van Helvoort and R. van de Molengraft, Realtime motion path generation using subtargets in a changing environment, *2006 American Control Conference* , 4243 (2006).
11. P. Estévez, J. Cappelletto, R. Acuña, F. Alvarez and G. Fernandez-López, Coordinated Navigation Using Dynamically Varying Velocity Fields (2007).
12. C. Perez-D'Arpino, W. Medina-Melendez, L. Fermin, J. Guzman, G. Fernandez-Lopez and J. C. Grieco, Dynamic Velocity Field Angle Generation for Obstacle Avoidance in Mobile Robots Using Hydrodynamics, *Advances in Artificial Intelligence - IBERAMIA 2008* , 372 (2008).
13. C. Laugier, D. Vasquez, M. Yguel, T. Fraichard and O. Aycard, Geometric and Bayesian models for safe navigation in dynamic environments, *Intelligent Service Robotics* **1**, 51 (2008).
14. Y. Tao, C. Faloutsos, D. Papadias and B. Liu, Prediction and indexing of moving objects with unknown motion patterns, *Proceedings of the 2004 ACM SIGMOD international conference on Management of data - SIGMOD '04* , p. 611 (2004).