Dynamical Path-Planning Algorithm of a Mobile Robot
Using Chaotic Neuron Model

Changkyu Choi, Sun-Gi Hong, Jin-Ho Shin, Il-Kwon Jeong and Ju-Jang Lee

Department of Electrical Engineering
Korea Advanced Institute of Science and Technology
373-1 Kusong-Dong Yusong-Gu, Taejon 305-701, Korea
Fax.: +82-42-869-3410 E-mail: flyers@iliad.kaist.ac.kr

Abstract
This paper describes a dynamical local path-planning algorithm of an autonomous mobile robot available for stationary obstacle avoidance using nonlinear friction. Dynamical path-planning algorithm is considered to accommodate the mobile robot to the dynamic situation of the path-planning nature. Together with the previous virtual force field (VFF) method, the path of the mobile robot is a solution of a path-planning equation. Local minima problems in stationary environments are solved by introducing nonlinear friction into the chaotic neuron. Because of the nonlinear friction, the proposed path-planner reveals chaotic dynamics in some parameter regions. This new path-planner is feasible to guide on real-time the mobile robot avoiding stationary obstacles and reaching the goal. Computer simulations are presented to show the effectiveness of the proposed algorithm.

1 INTRODUCTION

The path-planning problem can be classified into two categories. One considers only the path-planning, so it becomes an optimization problem, a global search problem, or a heuristic path-finding problem, etc. The other considers the path-planning and the dynamics of robots altogether[11]. A careful consideration of the dynamics of a mobile robot is, however, meaningless because the dynamics has not much to do with path-planning. The side effects caused by neglecting the dynamics of the mobile robot are overcome by a planned path tracking algorithm. Taking the velocity and/or the acceleration into account is just enough. For example, in the case of physical field approach such as virtual force field (VFF) method, it involves not only the position of the mobile robot but also the velocity and the acceleration of that in the path-planning algorithm. But authors developing their algorithms using VFF method did not take that point into account much.

In the proposed dynamical local path-planning algorithm, the robot’s path will be the solution of a continuous-time differential equation. The term “dynamical” results from this equation. The continuous-time differential equation is established based on Newton's second law,

\[ m \ddot{x} = F_t = F_{tr} + F_a = -\nabla V \]  

(1)

The total virtual force, \( F_t \), is thought to be a negative gradient of the electrostatic potential field. A virtual attractive force, \( F_a \), of constant magnitude is applied to the mobile robot, pulling it toward the target and all virtual repulsive forces in the circular active window are added up to yield the resultant repulsive force, \( F_r \). Applying the \( F_t \) into the eq.(1), however, the path-planner produces a spiral trajectory around the target. Thus, the attractive force must be modified to accommodate the robot to the additional information, i.e. the velocity and/or the acceleration.

Kuc et al.[10] and Borenstein et al.[2] mentioned the common drawbacks of ultrasonic sensors. A possible solution to cope with these shortcomings is to group the sensors into sensor suites. Focusing our attention to the polar obstacle density[3] and the index of performance[12], the repulsive forces will be calculated by the obstacle vector representation.

So far, the path-planning equation is always stable in the sense of Lyapunov. However, the robot cannot escape from the local minima in which it is trapped unless it has sufficient momentum. Tani[13] mentioned the sensitivity of the chaotic wandering robot to the initial condition. To escape from the local minima, the equilibrium point of null velocity is made to be unstable by substituting the nonlinear friction for the usual linear friction.

The organization of this paper is as follows. In section 2, the dynamical path-planning algorithm of a mobile robot based on Newton’s second law is considered. This algorithm works well for all cases but poses the local minima problem. To solve this problem, the linear friction is replaced by the time-varying nonlinear friction to cause the instability of the path-planning equation in section 3. Implementation of the nonlinear friction into a chaotic neuron is also discussed. Various simulation results are investigated at the end of section 2 and section 3. Finally, we draw some conclusions and discuss about the proposed
path-planner.

2 DYNAMICAL PATH-PLANNING ALGORITHM

2.1 Attractive Force Modification

As shown in Fig.1, the previous attractive force vector is represented as \( \mathbf{F}_{a} \). The attractive force from the robot center point toward the target is proper for a stationary situation. But robot is always moving and we don't assume that the robot moves at constant speed. Assuming that the mobile robot moves in the direction of \( \mathbf{F}_{v} \) which can be thought as a momentum, the attractive force should be modified to \( \mathbf{F}'_{a} \) considering the dynamical situation as shown in Fig.1, so that the momentum force \( \mathbf{F}_{v} \) and the new attractive force \( \mathbf{F}'_{a} \) are added to result in a previous attractive force \( \mathbf{F}_{a} \).

\[
\mathbf{F}_{a} = F_{ac}(\cos \theta_{a} \mathbf{i} + \sin \theta_{a} \mathbf{j})
\]

\[
\mathbf{F}_{v} = F_{cc} \mathbf{x}
\]

\[
\mathbf{F}'_{a} = \mathbf{F}_{a} - \mathbf{F}_{v}
\]

where \( \mathbf{i} \) and \( \mathbf{j} \) are the unit vectors, and \( \mathbf{x} \) is a two-dimensional velocity vector, \( \mathbf{F}_{ac} \) is the constant, and \( \theta_{a} \) is the angle of \( \mathbf{F}_{a} \). Thus, \( \mathbf{T}' \) and \( \mathbf{F}'_{a} \) are the new target and the new attractive force at that moment, respectively.

For the continuous-time differential equation used in our path-planner, the previous information (\( \mathbf{x} \)) about the mobile robot is reserved. Thus, the magnitude of attractive force is smaller than that of the stationary situation because of the momentum force, \( \mathbf{F}_{v} \).

2.2 Repulsive Force Modification: Obstacle Vector Representation

There are remedies to overcome the shortcomings of the ultrasonic sensors. One of the remedies is to group the sensors into sensor suites. However, since this method is based on the fire-scheduling, it's not an inherent cure for the shortcomings of ultrasonic sensors. The repulsive forces will be calculated by the proposed obstacle vector representation[5]. The magnitude and the angle of an obstacle vector is the possibility of the existence of obstacles at that distance and at that direction, respectively. The procedure to calculate obstacle vectors is based on the polar obstacle density and the index of performance[5].

Hence, the path-planner can be expressed as the following single equation.

\[
m\ddot{x} + d \dot{x} = m \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} + d \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \mathbf{F}_{t} = \mathbf{F}_{a}' + \mathbf{F}_{r}'
\]

Since the robot's path, \( (x_{p},y_{p}) \) is the solution of the above continuous-time differential equation, it is at least in \( C^{2} \) where \( C^{n} \) denotes a set of \( n \)-times continuously differentiable functions. Therefore, the planned path is smooth enough to steer the robot toward the target. This is one of the reasons why we use the path-planner as a continuous-time differential equation.

2.3 Simulation Results

A spiral trajectory around the target which is a drawback of the previous method is shown in Fig.2. The effects of the new attractive force alone and both of the new attractive and repulsive forces are shown in Fig.3. (a) and (b), respectively. The proposed dynamical path-planning algorithm works well for all cases except for the local minima problem.

3 LOCAL MINIMA PROBLEM

The VFF method is a type of steepest descent method. If the gradient happens to be zero (i.e. the virtual force is zero and remains zero thereafter), then the robot cannot move and the velocity of the robot becomes zero, for the origin of the velocity-acceleration space is a stable equilibrium point of eq.(5). In this case, the robot fails to find the target. This is the local minima problem. In order to bring the robot back to its job of finding the goal, either the local minima should be avoided or the robot should escape from it. The former is mentioned in Kim et. al.[8] and is suitable for global path-planning. The latter is another possible method[2][4], escaping from the local minima by making the equilibrium point of null velocity unstable.

3.1 Introducing time-varying nonlinear friction, \( fr \)

Friction is relevant to velocity. In eq.(5), the friction is proportional to the velocity of the robot. As a result of this fact, the proposed path-planning equation is always stable in the sense of Lyapunov. However, the robot cannot escape from the local minima in which it is trapped unless it has sufficient momentum. The ability to pass through the local minima depends on the height of the potential barrier around it and the momentum of the robot. Thus, the momentum method cannot work well in an arbitrary environment.

Consider the following time-varying nonlinear friction which is a function of the velocity and time.

\[
fr(\dot{x}, wt) = d_{1} \sin (wt) \dot{x} + d_{2} \dot{x}^{3}
\]

For a half cycle, \( i.e. \dot{x} \times fr(\dot{x}, wt) \geq 0 \) the nonlinear friction lies only on the 1st and 3rd quadrants. In these phases, the origin is the only equilibrium point and is stable in the sense of Lyapunov. For the other phases, when \( \dot{x} \times fr(\dot{x}, wt) < 0 \), three equilibrium points exist. One is the origin and the other two are symmetrically placed about the origin in \( \dot{x} \) axis and are time-varying. The origin becomes unstable and the others are stable. A role of the nonlinear friction is to make the velocity of the robot a non-zero value. The sign of the initial velocity when \( fr(\dot{x}, wt) \) changes its sign determines the sign of the time-varying equilibrium point, which the velocity of the robot tends to track. The maximum magnitude of the time-varying equilibrium points relies on the parameters, \( d_{1} \) and \( d_{2} \). This maximum velocity should not exceed the motor driver's limitation. After setting this speed, the only design problem is how to choose \( w \), which, however, is chosen heuristically.
The path-planner is amended to the following:

\[ m\ddot{x} + d_1\sin(\omega t)\dot{x} + d_2\dot{x}^3 = m \left[ \frac{\dot{x}}{y} \right] + d_1\sin(\omega t) \left[ \frac{\dot{y}}{y} \right] + d_2 \left[ \frac{\dot{y}^3}{y^3} \right] \]

\[ \mathbf{F}_t = \mathbf{F}_a' + \mathbf{F}_r' \]  

(7)

Introducing the time-varying nonlinear friction adds instability into the path-planner. The more unstable the path-planner, the higher the potential energy of the mobile robot. If the increased potential energy can overcome the barrier around the local minima, the robot can escape from that local minima. Hence, the path-planner, eq.(7) can generate a path to solve the local minima problem.

3.2 Analysis of the Path-Planner, eq.(13)

Consider an external force that can approximate the virtual force relies on the environment, the closed form solution of virtual force is hard to find. Even if a solution exists, a wide variety of environments must be tested. For simplicity, we consider an obstacle moving periodically in front of the robot provided that the mobile robot is working. Since the virtual force is hard to find. Even if a solution exists, a wide variety of environments must be tested. For simplicity, we consider an obstacle moving periodically in front of the robot provided that the mobile robot can move only back and forth. Thus, eq.(7) can be modified as follows with the assumption of a unit mass.

\[ \ddot{x} + d_1\sin(\omega t)\dot{x} + d_2\dot{x}^3 = \frac{1}{\cos^2(\omega t - x)^2} \]  

(8)

where \( \omega = 0.04 \) and \( \Omega = \sqrt{2} \). Two incommensurate frequencies represent the assumption that the dynamics of path-planner and the variation of environment are not related, so that these do not make a resonance.

Considering the path-planning equation in (8), the external force and the time-dependent term are eliminated, which give an additional degree of freedom to the equation.

\[ \ddot{x} + \alpha \dot{x} + \beta \dot{x}^3 = 0 \]  

(9)

Differentiating the above equation yields

\[ \ddot{v} - \alpha^2v^2 - 4\alpha\beta v^3 - 3\beta^2v^5 = 0 \]  

(10)

where \( \dot{v} \) is the velocity of the robot, \( \ddot{v} = v \).

We focus our attention to eq.(10). Considering only the eq.(10), \( \dot{v} \) and \( \ddot{v} \) possess information about the position and the acceleration, respectively. Thus, potential function of eq.(10) is calculated based on the following.

\[ F = m\ddot{v} = -\nabla V \]  

(11)

After some calculation, the potential function \( V \) is

\[ V(v) = -\frac{1}{2}\alpha^2v^2 - \alpha\beta v^4 - \frac{1}{2}\beta^2v^6 \]  

(12)

and the solutions of \( V = 0 \) are \( v^2 = 0 \) in single root and \( v^2 = -\alpha/\beta \) in double roots. Hence, eq.(10) exhibits the resemblance of the double well potential problem if \( v \) satisfies the following conditions:

\[ \alpha \beta < 0 \quad \text{and} \quad -\sqrt{-\frac{\alpha}{\beta}} < v < -\frac{\alpha}{\beta} \]  

(13)

If \( v \) does not meet the above conditions, eq.(10) diverges to positive or negative infinity. Coming back to the eq.(8), it reveals the characteristics of the double well potential[6] for the variables of \( \dot{x} \) and \( \ddot{x} \) (or \( \dot{v} \) and \( \ddot{v} \)) in the case that the external force and the perturbation exist.

3.3 Implementation of Nonlinear Friction into the Chaotic Neuron Model

Aihara et. al.[1] suggest a chaotic neuron model based upon Caianiello's neuron equation. The chaotic neuron model which includes conventional models of neural networks reads

\[ p(n + 1) = f(A(n) - \alpha \sum_{r=0}^{n} k^r g(p(n - r)) - \delta) \]  

(14)

Defining the internal state \( q(n + 1) \) by

\[ q(n + 1) = A(n) - \alpha \sum_{r=0}^{n} k^r g(p(n - r)) - \delta \]  

(15)

reduces eq.(14) to the following equations,

\[ p(n + 1) = f(q(n + 1)) \]  

(16)

\[ q(n + 1) = kq(n) - \alpha g(f(q(n))) + a(n) \]  

(17)

\[ a(n) = A(n) - kA(n - 1) - (1 - k)\delta \]  

(18)

Since eq.(7) is a time-varying nonlinear differential equation, the corresponding one in discrete-time domain cannot be obtained. Gaining the underlying philosophy in eq.(13), we can approximate it utilizing a chaotic neuron, of which structure is basically discrete. If the output function, \( f \) and the refractory function, \( g \) take the following forms (Fig.4):

\[ f(q) = \frac{\rho}{1 + e^{-q/\Phi}} - \frac{\rho}{2} \]  

(19)

\[ g(p) = \rho f(p + \eta) - f(p - \eta) \]  

(20)

where \( \Phi \) and \( \eta \) controls the shape of the functions \( f \) and \( g \), respectively, and \( \rho \) the magnitude of the two functions, then the eq.(17) resembles the philosophy of eq.(7) like the following manner:

\[ m\ddot{x} = p(n + 1) = \begin{bmatrix} p_x(n + 1) \\ p_y(n + 1) \end{bmatrix} \]  

with \( a(n) = \begin{bmatrix} F_x - kF_x - (1 - k)\delta \\ F_y - kF_y - (1 - k)\delta \end{bmatrix} \)  

(21)

where \( p_x(n + 1) \) and \( p_y(n + 1) \) are outputs of the two independent chaotic neurons. The bifurcation diagram of \( q(n + 1) \) is shown in fig.5.
Casting a glance at eq.(21), it looks somewhat strange because the left-side of it is a continuous-time equation while the right-side of it is discrete-time one. It is obvious, however, that the external force, \( F_e \) is calculated at every sampling time intervals \( T \), whereas the discrete-time index 1 corresponds to the forward propagation time \( \tau \) of the chaotic neuron with \( \tau \ll T \). Therefore, the right-side of eq.(21) is not a discrete-time signal but a piecewise continuous-time one. As seen in eq.(21), the output \( p_e(n+1) \) (or \( p_p(n+1) \)) of the chaotic neuron manipulates the acceleration, \( \dot{x} \) (or \( \ddot{y} \)) of the mobile robot. The output function \( f_e \) (or \( f_p \)) represents physical limitations of the maximum and minimum acceleration rate, for \( f_e \) (or \( f_p \)) takes the value in \([-p/2, p/2] \). The refractory function, \( g_x \) (or \( g_y \)) is symmetric about origin at which it takes the maximum value, \( |p| \) and the value of it goes to 0 as the argument of it approaches to positive or negative infinity. Because of the very shape of refractory function, the velocity of a robot cannot be 0, so that the robot can be prevented from a local minima situation. The amount of threshold \( \delta \times (1 - k) \) is subtracted from \( a_x(n) \) (or \( a_y(n) \)) to cover the shortcoming of sensor misreadings.

### 3.4 Simulation Results

The proposed algorithm, eq.(21) (or eq.(7)) remedies the shortcoming of eq.(5). Path-planner, eq.(21) (or eq.(7)) can guide a mobile robot in avoiding stationary obstacles, escaping from local minima and finally reaching the goal as shown in Fig.6. Once the robot is trapped in the local minima, and as time goes on, the robot gains enough energy to escape from the local minima due to the diverging chaotic dynamics of the path-planner.

To simplify the eq.(7), we consider an obstacle moving periodically in front of the robot provided that the mobile robot can move only back and forth. The eq.(8) is then simulated as shown in Fig.7 for some parameter set \( (d_1, d_2) \). The largest value of \( d_2 \) is near 25 in order not to make eq.(8) diverge. Fixing \( d_2 \) in the vicinity of this value and varying \( d_1 \), Poincare sections \( (s(t) = 0) \) are investigated. Fig.7 shows a strange attractor. We suspect that the direction of \( \dot{x} \) near the origin is the direction of stable manifold and the other is of unstable one. Hence, eq.(8), having a saddle point at the origin, reveals chaotic dynamics in some parameter regions.

### 3.5 Goal-Positioning Algorithm

The robot’s speed is decreased and increased at frequency \( w \) repeatedly because the dynamics of path-planner, eq.(7) toggles its state from a stable phase to an unstable phase, and vice versa. The unstable diverging phase keeps the mobile robot from trapping the local minima, while the stable phase leads the robot to the target in a steepest descent manner. Not having embedded any intelligence in the proposed path-planner, the robot finds the target ineffectively when the path-planner is equipped with the unstable phase in a local minima-free environment. In this case shown in Fig. 6, the robot finds the target successfully but not effectively. To avoid this kind of useless path-planning, an additional strategy to reach the target, called goal-positioning algorithm must be considered as in [7].

Consider an additional force called goal-positioning force, \( F_{GP} \).

\[
F_{GP} = s(t) \times F^* = s(t)[-m(2\zeta w_n\dot{x} + \omega_n^2(x - x_d)) - p(n+1)]
\]

where \( s(t) \) is a type of switching function, \( \zeta \) is a damping ratio, \( w_n \) is a natural undamped frequency and \( x_d \) is the target position. If the right side of the eq.(21) is modified to \( p(n+1) + F_{GP} \) and \( s(t) \) is set to be 1, then the path-planner finally becomes

\[
\ddot{x} + 2\zeta w_n\dot{x} + \omega_n^2(x - x_d) = 0
\]

From the linear system theory, it is clear that both \( \ddot{x} \) and \( \dot{x} \) go to zero and \( x \) goes to \( x_d \). If the parameters \( \zeta \) is chosen to be 1, the eigenvalues of eq.(23) are double roots at \( -w_n \) so that \( (\ddot{x}, \dot{x}, x) \) converges exponentially to \((0, 0, x_d)\).

### 4 CONCLUSIONS

A dynamical local path-planning algorithm for stationary obstacle avoidance is proposed. We modify the attractive force to accommodate the mobile robot to the dynamical path-planning algorithm by introducing the momentum force \( F_{v} \). To overcome the usual drawbacks of ultrasonic sensors, the repulsive force is amended to incorporate concept of the obstacle vector representation.

In conclusion, several notes are addressed. First, the trajectory that the robot must follow is the solution of the continuous-time differential equation of the second order. Therefore, the generated path is continuously differentiable at least two times. Second, since the resultant virtual force acts as an external input of the path-planner, we can control the acceleration and deceleration by adjusting the parameters in the equation properly. This is a physically acceptable notion because the path-planner is established based on the Newton’s second law. Third, as it was already pointed out in the literature[9], the ratio between attractive and repulsive forces are suitably chosen. Furthermore, the parameter values of the path-planner must be selected in consideration of the physical limitations. These are under our investigations. Fourth, by utilizing the time-varying nonlinear friction, the local minima problem is solved by replacing linear friction by a nonlinear friction. The nonlinear friction causes instability to the path-planning equation and the instability enables a mobile robot to escape from local minima. Based on the numerical analysis, we show that chaotic dynamics are embedded in the proposed algorithm. Hence, the use of intentional chaotic dynamics might improve the performance of the system of interest.

### References


Fig. 4 Functions used in chaotic neuron model.

Fig. 6 Planned trajectory that can escape from local minima (eq. (21)).

Fig. 5 Bifurcation diagram of $q(n+1)$.

Fig. 7 Poincare section of eq. (7) ($d_1=400, d_2=25.002$)