Wheeled robot navigation based on a unimodal potential function

Gregor Klančar
Faculty of Electrical Engineering
University of Ljubljana
Ljubljana, Slovenia
gregor.klancar@fe.uni-lj.si

Gašper Mušič
Faculty of Electrical Engineering
University of Ljubljana
Ljubljana, Slovenia
gasper.music@fe.uni-lj.si

Hao Chen
School of Electrical and Power Engineering
China University of Mining and Technology
Xuzhou, China
hchen@cumt.edu.cn

Marija Seder
Faculty of Electrical Engineering and Computing
University of Zagreb
Zagreb, Croatia
marija.seder@fer.hr

***Abstract*—This article proposes wheeled mobile robot navigation based on a model predictive control over unimodal potential function. The navigation potential function has a single minimum and no local minima. It is obtained from a discrete graph-based search algorithm that can update the environment changes. Obtained navigation function enables continuous convergent navigation where gradient direction is computed to steer the robot towards the goal. Model predictive control using particle swarm optimization is applied to find the path within areas of the lowest potential and considering the control effort. The proposed solutions feature simple implementation, at moderate computational effort and with good navigation results.**

***Index Terms*—Navigation, potential field, motion control, model predictive control, path planning.**

I. Introduction

Path planning and motion control are basic operations of any autonomous mobile robot and have been extensively studied in the literature [1], [2], [3]. In these tasks robot has to move safely between obstacles from the initial to the desired goal pose. This is usually done by computing the reference path in the first step and then applying some control algorithm to follow this path. In case of initial pose error, transient control error or any disturbances in robot operation, the precomputed reference path may no longer be the preferred one. An even more obvious example is a change of the environment which would require path re-planning first. Therefore path planning and motion control should better be merged to a joint action called navigation [1], [4], [5]. The navigation function is precomputed and should also have fast update capabilities to cope with possible environment changes. The navigation function can then be used to control the robot from any location towards the goal. One can also apply model predictive approaches (e.g. [6]) over navigation function to find optimal control actions which consider robot motion model as well as environment presentation.

This work was performed in the frame of Scientific and Technological Cooperation between Chinese and Slovenian Governments 2018-2020.

Basic potential field path planning approaches can be used for path pre-planning and for on-line navigation where control actions are computed to drive a vehicle in the direction of the steepest descent of the potential field (navigation function). The potential navigation function can be used in a path optimization as done in [7] where particle swarm optimization is applied. However most of such basic navigation functions suffer from local minima due to concave obstacle shapes where robot can become trapped. There are several approaches to prevent the trapped situations. One can simply modify map of the environment to present concave obstacles by convex presentations [8] or use adaptive potential field generated using multiple attractions points instead of only one in the goal [9]. It is also possible to modify the potential field in unstable equilibrium by introducing perturbations of the field [10] or adding virtual obstacles to repel the robot from the local minima [11] and similar solutions [12].

The main novelty of this work is construction of the navigation function which only has one global minimum and no local minima and is integrated in a predictive control using particle swarm optimization. The navigation function is obtained from known environment map and graph search algorithm D^* (or A^* in static maps), similarly as done in [4] and [5]. To obtain continuous potential function the area of each discrete cell is split to triangles to obtain continuous transition of potential between the triangle's planes. The obtained potential function is strictly decreasing, which means that every position that is closer or more optimal has smaller potential value than the others. The smallest potential is in the goal point and the largest is on the obstacles. Based on the obtained interpolated navigation function two control algorithms are provided, the gradient-following control and model-based predictive control (MPC) using particle swarm optimization. The gradient-

978-1-7281-1374-6/19/$31.00 ©2019 IEEE

following control is simply derived from the interpolated potential function and used as a reference driving direction in orientation control. While the MPC searches continuous space of bounded control actions to find control set for a given horizon. Inside horizon circular robot motion is supposed which lowers the dimension of parameter space and makes the optimization computationally more effective at similar performance. To obtain feasible controls also bounds on maximal velocities and accelerations are considered. Performance of the proposed robot navigation is illustrated on several examples.

II. Potential field navigation function

The navigation function has the lowest potential in the desired goal state and the potential increases for poses away from the goal and has the highest values at the obstacles. Knowing the environment layout (e.g. Fig. 1) the navigation function can be defined by a smooth mapping from environment to potential field based on distances to the goal and to the obstacles. This approach is known to be computationally very effective but has several local minima and therefore the goal (global minimum of potential) may not always be found. In Fig. 1 the search can easily get trapped inside the obstacle (suppose starting position $x = 5$ m, $y = 4$ m and goal at $x = 4.25$ m, $y = 8.25$ m).

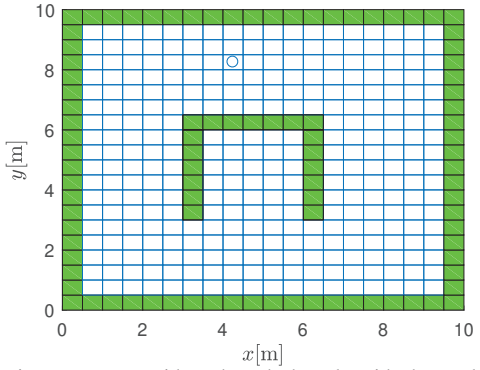

Fig. 1. Environment map with u-shaped obstacle with the goal location at $x = 4.25$ m and $y = 8.25$ m.

Example of potential field computed from D^* discrete search algorithm is shown in Fig. 2. The discrete function in

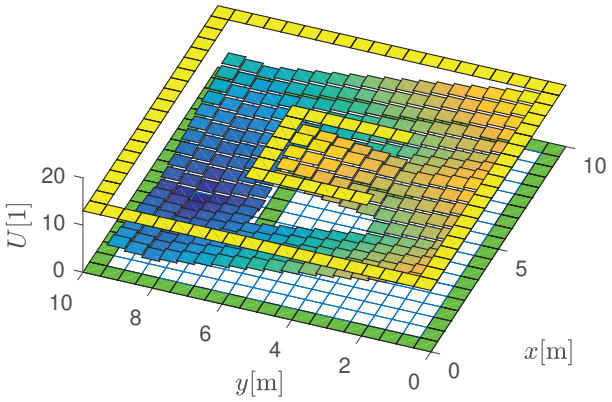

Fig. 2. Discrete potential field U obtained from the graph search algorithm with cell size 0.5 m. Goal location at $x = 4.25$ m , $y = 8.25$ m has the lowest (zero) potential.

Fig. 2 is unimodal with the only minimum at the goal cell and all the other cells potentials increase from there. The value of potential in each cell represents the distance to the goal cell computed as the sum of distances between the cells along the path. In the graph search 4-neighbour search expansion is used. Before applying this function to a control algorithm it needs to be made continuous so that unique potential value is obtained for any position. This can be done effectively by a simplex interpolation function, similarly as done in [5].

Each cell is presented by eight triangles with the common vertex at the cell center $\mathbf{i} = [x_i, y_i]^T$ with potential $P_i = U(\mathbf{i})$ that is already computed. The other vertices of triangles are defined by the cell vertices $\mathbf{v} = [x_v, y_v]^T$ and the middle points $\mathbf{m} = [x_m, y_m]^T$ on the cell's borders, as shown in Fig. 3.

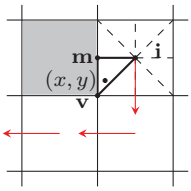

Fig. 3. Division of a cell into eight triangles; the cost at (x, y) is interpolated between costs at vertices $\mathbf{v}$, $\mathbf{m}$, and $\mathbf{i}$, where only at $\mathbf{i}$ the cost is calculated by the path search algorithm, and the path is noted by arrows.

For a given point inside the triangle's plane the absolute barycentric coordinates (λ_i, λ_v, λ_m) are computed considering

$$\begin{aligned} x &= \lambda_i x_i + \lambda_v x_v + \lambda_m x_m \\ y &= \lambda_i y_i + \lambda_v y_v + \lambda_m y_m \\ \lambda_i &+ \lambda_v + \lambda_m = 1 \end{aligned} \tag{1}$$

from where the coefficients are

$$\begin{aligned} \lambda_i &= \frac{x y_v - x_v y - x y_m + x_m y + x_v y_m - x_m y_v}{x_i y_v - x_v y_i - x_1 y_m + x_m y_i + x_v y_m - x_m y_v} \\ \lambda_v &= \frac{x y_i - x_i y - x y_m + x_m y + x_i y_m - x_m y_i}{x_i y_v - x_v y_i - x_1 y_m + x_m y_i + x_v y_m - x_m y_v} \\ \lambda_m &= 1 - \lambda_i - \lambda_v \end{aligned} \tag{2}$$

Then the potential $P(x, y)$ in the point x, y is interpolated using coefficients in (2)

$$\begin{aligned} P(x,y) &= \lambda_i P_i + \lambda_v P_v + \lambda_m P_m \\ P_v &= 0.25 \textstyle\sum_{\mathbf{j}} U(\mathbf{j}) \quad ; \quad ||\mathbf{j} - \mathbf{v}|| = \frac{d_c}{\sqrt{2}} \\ P_m &= 0.5 \textstyle\sum_{\mathbf{j}} U(\mathbf{j}) \quad ; \quad ||\mathbf{j} - \mathbf{m}|| = \frac{d_c}{2} \end{aligned} \tag{3}$$

where d_c is the length of the cell edge and $\mathbf{j}$ is the center point of one of the surrounding cells. P_v is potential in point $\mathbf{v}$ computed as the mean potential of the four cells that have common vertex $\mathbf{v}$ and P_m is potential in point $\mathbf{m}$ obtained as a mean potential of the two cells sharing the same edge with point $\mathbf{m}$.

The obtained continuous potential function has a single global minimum at the goal position as seen in Fig. 4 to which robot can arrive from any other point in the environment optimally without collision and with nonzero descent.

III. Control law

Control law needs to move the robot towards the steepest descent of the potential field. This is achieved by the gradient following controller and model-based predictive controller as explained in the sequel.

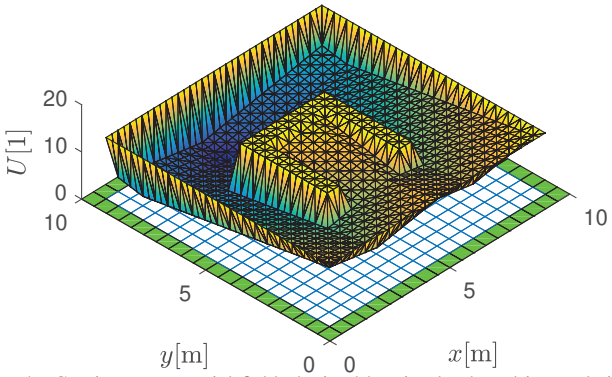

Fig. 4. Continuous potential field obtained by simplex-based interpolation.

A. Gradient following control

The potential function (3) is strictly decreasing, meaning that every position that is closer or more optimal has a smaller potential value than the others. To drive towards the goal the optimal driving direction $\mathbf{s} = [s_x, s_y]^T$ is a gradient $\mathbf{s} = -\left[\frac{\partial P(x,y)}{\partial x}, \frac{\partial P(x,y)}{\partial y}\right]^T$ computed as follows

$$\mathbf{s} = \begin{bmatrix} -\frac{P_i y_v - P_v y_i - P_i y_m + P_m y_i + P_v y_m - P_m y_v}{x_i y_v - x_v y_i - x_i y_m + x_m y_i + x_v y_m - x_m y_v} \\ \frac{P_i x_v - P_v x_i - P_i x_m + P_m x_i + P_v x_m - P_m x_v}{x_i y_v - x_v y_i - x_i y_m + x_m y_i + x_v y_m - x_m y_v} \end{bmatrix} \tag{4}$$

The reference driving direction is obtained in current robot position. Then, control actions can be computed based on orientation error and robot kinematics. Supposing differential-drive robot with current pose $\mathbf{c} = [x(t), y(t)]^T$, $\varphi(t)$, the reference orientation φ_r, the orientation error $e_\varphi(t)$ and the control actions ($v(t)$, $\omega(t)$) are as follows

$$\begin{array}{ll} \varphi_r = \arctan(\frac{s_y}{s_x}) & e_\varphi(t) = \varphi_r - \varphi(t) \\ \omega(t) = K_\omega e_\varphi(t) & v(t) = K_v ||\mathbf{g} - \mathbf{c}|| \end{array} \tag{5}$$

where K_ω, K_v are positive constants, $\mathbf{g} = [x_g, y_g]^T$ is the goal position and $\omega(t)$, $v(t)$ are the angular and the translational velocity, respectively. To obtain feasible actions v^*, ω^* that preserve current curvature, the maximal velocities (v_{MAX}, ω_{MAX}) and the maximal accelerations (a_{MAX}, α_{MAX}) are considered as follows

$$v'(t) = \begin{cases} v(t) & ; \quad a \le a_{MAX} \\ v(t - T_s) + \mathrm{sgn}(a) a_{MAX} T_s & ; \quad a > a_{MAX} \end{cases} \tag{6}$$

$$\omega'(t) = \begin{cases} \omega(t) & ; \quad |\alpha| \le \alpha_{MAX} \\ \omega(t - T_s) + \mathrm{sgn}(\alpha) \alpha_{MAX} T_s & ; \quad |\alpha| > \alpha_{MAX} \end{cases} \tag{7}$$

$$v^*(t) = \begin{cases} v'(t) & ; \quad \rho = 1 \\ v'(t)\,\mathrm{sgn}(v(t)) v_{MAX} & ; \quad \rho = \frac{|v'|}{v_{MAX}} \\ \frac{v'(t)}{\rho} & ; \quad \rho = \frac{|\omega'|}{\omega'_{MAX}} \end{cases} \tag{8}$$

$$\omega^*(t) = \begin{cases} \omega'(t) & ; \quad \rho = 1 \\ \omega'(t)\,\mathrm{sgn}(\omega'(t)) \omega_{MAX} & ; \quad \rho = \frac{|\omega'|}{\omega_{MAX}} \\ \frac{\omega'|(t)}{\rho} & ; \quad \rho = \frac{|v'|}{v_{MAX}} \end{cases} \tag{9}$$

where T_s is sample time, $a = \frac{v(t) - v(t - T_s)}{T_s}$, $\alpha = \frac{\omega(t) - \omega(t - T_s)}{T_s}$ and $\rho = \max(\frac{|v'|}{v_{MAX}}, \frac{|\omega'|}{\omega_{MAX}}, 1)$.

Obtained behavior of the gradient-based control is shown in the map in Fig. 5 based on the navigation function in Fig. 4. Control algorithm is very simple and fast to compute and guarantees to return a feasible action in every moment. Having a closer observation an oscillating behavior is observed at some of the obstacle borders. This oscillations can be influenced by selecting the controller gains, sample time and allowed accelerations. They appear near the obstacles where the navigation function has slope changes and consequently the gradient is switching its direction. Nevertheless the navigation still drives the robot safely towards the goal. To overcome

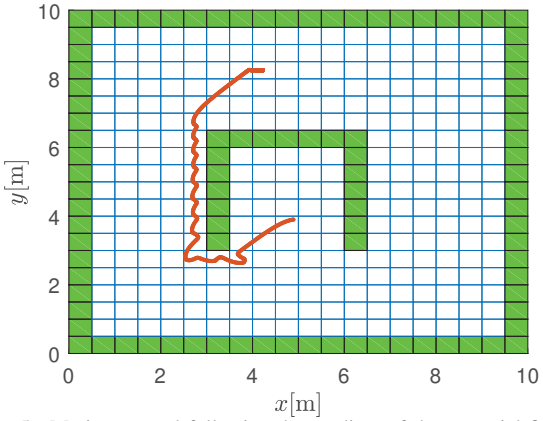

Fig. 5. Motion control following the gradient of the potential field.

this oscillating behavior a model predictive control algorithm is implemented.

B. Model-Based Predictive Control

Control actions are found by means of optimization where minimum of objective function is searched over the horizon h. The function reads

$$\begin{aligned} J(t + hT_s) = \min_{\mathbf{u}(t)} \textstyle\sum_{i=1}^{h} P(x(t + iT_s), y(t + iT_s)) + \\ \xi |e_\varphi(t + iT_s)| + \mathbf{u}^T(t) \mathbf{R} \mathbf{u}(t) \end{aligned} \tag{10}$$

where $\mathbf{u}(t) = [v_c, \omega_c]$, $\xi > 0$, $e_\varphi(t + iT_s) = \varphi_r(t + iT_s) - \varphi(t + iT_s)$, $\varphi_r(t + iT_s)$ is the orientation of the gradient at the coordinates $x(t + iT_s), y(t + iT_s)$ and $\mathbf{R}$ is the input weighting matrix. During horizon interval the robot is supposed to drive on a circular arc, therefore $v(t + T_s i) = v_c$, $\omega(t + T_s i) = \omega_c$, for $i \in 1 \cdots h$. This simplifies the optimization problem and lowers computational complexity. Control actions are constrained considering (6)-(9) during optimization which produces smoother robot motion and extends the lifetime of electric motor due to lower vibrations similarly as in the case of analyzed asymmetric construction gap in reluctance motors [13].

Robot pose during horizon interval is predicted using kinematic model

$$\begin{aligned} x(t + T_s) &= x(t) + v(t) T_s \cos\left(\varphi(t) + \tfrac{\omega(t) T_s}{2}\right) \\ y(t + T_s) &= y(t) + v(t) T_s \sin\left(\varphi(t) + \tfrac{\omega(t) T_s}{2}\right) \\ \varphi(t + T_s) &= \varphi(t) + \omega(t) T_s \end{aligned} \tag{11}$$

Particle Swarm Optimization (PSO) is chosen to solve the minimization problem. In PSO a group of particles is used

to explore the parameter space of the minimization function. Each particle k is characterized by parameter vector $\mathbf{p}_k$ defining its position in the parameter space and increment vector $\Delta\mathbf{p}_k$ defining the change within the parameter space. In the case of this work the parameter space represents robot control actions ($\mathbf{p}_k = [v_k, \omega_k]^T$). During optimization population of all candidate solutions are updated according to objective function that defines a measure of quality. Each particle keeps track of its parameters and remembers its best parameters $\mathbf{pB}_k$ achieved so far together with associated objective function $J_k = f(\mathbf{pB}_k)$. During optimization also the best parameter vector obtained so far for entire swarm $\mathbf{gB}$ is stored.

In each control-loop iteration particles update mimic cognitive and social behavior by means of the following rules

$$\begin{aligned}\Delta\mathbf{p}_k &\leftarrow \gamma\Delta\mathbf{p}_k + c_1\underset{(0,1)}{\mathbf{rand}}(\mathbf{pB}_k - \mathbf{p}_k) + c_2\underset{(0,1)}{\mathbf{rand}}(\mathbf{gB} - \mathbf{p}_k)\\ \mathbf{p}_k &\leftarrow \mathbf{p}_k + \Delta\mathbf{p}_k\end{aligned} \tag{12}$$

where γ is the inertia factor, c_1 is self-cognitive constant and c_2 is social constant. In addition $\mathbf{rand}_{(0,1)}$ is a vector of uniformly distributed values in range $(0, 1)$. The dimensions of vectors in (12) equal the dimension of the parameter search space. The parameters γ, c_1 and c_2 are positive tuning parameters.

Obtained results using model-based predictive control using PSO are shown in Figs. 6 and 7. In optimization the prediction horizon is set to $h = 20$, sampling time is $T_s = 0.033$ s, maximal velocities $v_{MAX} = 1$ m/s, $\omega_{MAX} = 6$ 1/s and maximal accelerations $a_{MAX} = 1$ m/s^2, $\alpha_{MAX} = 6$ 1/s^2. PSO-ralated parameters are as follows: number of particles is set to 25, number of iterations in each sample time is 20, inertia factor is $\gamma = 0.8$, cognitive factors are $c_1 = 0.5$ and $c_2 = 0.5$, respectively.

Constrains on v_{MAX}, ω_{MAX}, a_{MAX} and α_{MAX} are in PSO introduced by adding a penalty function $g(\mathbf{p}_k)$ to the MPC objective function (10). At each particle evaluation its overall objective function reads $F_k(\mathbf{p}_k, t) = J(t + hT_s) + \sum_{j=1}^{m} K_j \max(0, g_j(\mathbf{p}_k))$ where m is the number of constraints, $K_j > 0$ and $g_j(\mathbf{p}_k) \leq$ are the inequality constraints (e.g. for $j = 1$ the constraint is $|v_k| \leq v_{MAX}$ and the relative violations is $g_1(\mathbf{p}_k) = |v_k| - v_{MAX}$).

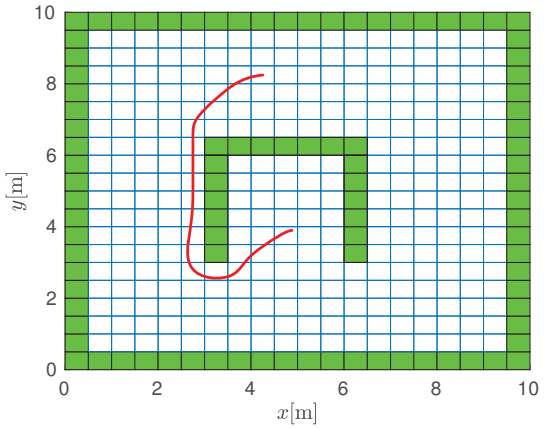

Fig. 6. Model predictive control with PSO on interpolated potential function.

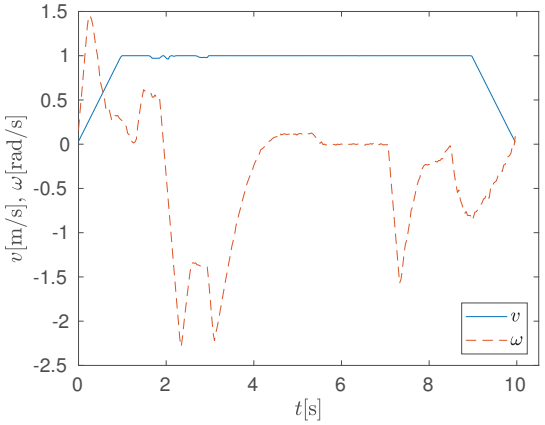

Fig. 7. Control actions obtained during MPC Particle Swarm Optimization considering constraints on v_{MAX}, ω_{MAX}, a_{MAX} and α_{MAX}.

The proposed navigation function with model predictive control can reliably find the path towards the goal location as shown in the examples. The obtained path is much smoother as in case of the simpler gradient-direction based control. The obtained control is always feasible as it adapts on-line to disturbances and considers constraints on maximal velocities and accelerations.

C. Comparison to Fix Set Optimization

The proposed control method based on Particle Swarm Optimization is compared to the MPC based on Fix Set Optimization developed in [5]. For comparison, the same objective function (10) is used for optimization of trajectories. The control actions $u(t)$ are searched with respect to the fix set of control actions U_{FSO} determined each time step. The fix set U_{FSO} is composed of the current velocity v_c, ω_c and lower and upper velocities according to kinematic and dynamic constraints as follows:

$$U_{FSO} \leftarrow \left\{ u = \begin{bmatrix} v \\ \omega \end{bmatrix} \;\middle|\; \begin{array}{l} v \in \{v_l, v_c, v_u\} \\ \omega \in \{\omega_l, \omega_c, \omega_u\} \end{array} \right\}, \tag{13}$$

where

$$\begin{aligned} v_l &\leftarrow \max\{v_c - a_{MAX}T_s, 0\} \\ \omega_l &\leftarrow \max\{\omega_c - \alpha_{MAX}T_s, -\omega_{MAX}\} \\ v_u &\leftarrow \min\{v_c + a_{MAX}T_s, v_{MAX}\} \\ \omega_u &\leftarrow \min\{\omega_c + \alpha_{MAX}T_s, \omega_{MAX}\}, \end{aligned} \tag{14}$$

which gives 3×3 combinations of velocity pairs v, ω. To simplify the optimization problem robot is supposed to drive on a circular arc of radius $\frac{v}{\omega}$, where first part of trajectory has constant velocity $u = [v, \omega] \in U_{FSO}$, while the second part of trajectory has linearly decreasing velocities with zero at the time $t + T_s h$:

$$u(t + T_s i) \leftarrow \begin{cases} u & |i \leq h - T_{dec}, \\ u\dfrac{h - i}{T_{dec}} & |h - T_{dec} < i \leq h, \end{cases} \tag{15}$$

where T_{dec} is a number of steps to slow down from the chosen velocity $u = [v, \omega]$:

$$T_{dec} \leftarrow \max\left\{ \left\lceil \frac{v}{a_{MAX}T_s} \right\rceil, \left\lceil \frac{|\omega|}{\alpha_{MAX}T_s} \right\rceil \right\}. \tag{16}$$

Robot pose during horizon interval is predicted using kinematic model (11). This procedure creates nine possible trajectories. Among them, only feasible ones that do not collide with the obstacle are considered in the optimization. This is simply checked by comparing the value of the navigation function P in each time step of trajectory during horizon interval. Then, the best feasible trajectory is chosen that has the lowest value of the objective function (10) and the first control action of the best trajectory is applied. At the next time step, the new fix set of control actions is determined and the same procedure is repeated. Slowing down to zero velocity at the horizon is necessary to ensure feasible trajectories in each time step.

The obtained velocity profile for the same obstacle configuration and constraints on velocities is shown in Fig. 8. The

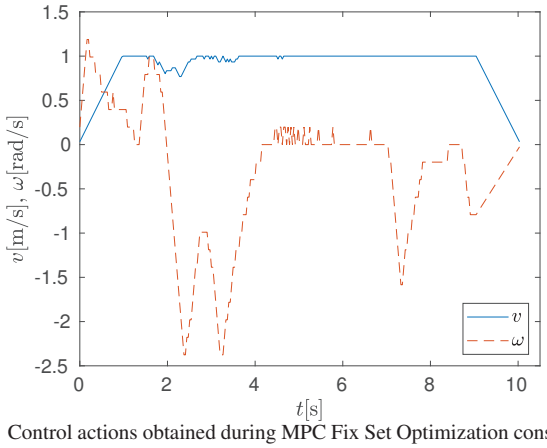

Fig. 8. Control actions obtained during MPC Fix Set Optimization considering constraints on v_{MAX}, ω_{MAX}, a_{MAX} and α_{MAX}.

comparison of trajectories obtained by the proposed control method (MPC-PSO) and the Fix Set Optimization (MPC-FSO) is shown in Fig. 9. It can be seen that much smoother

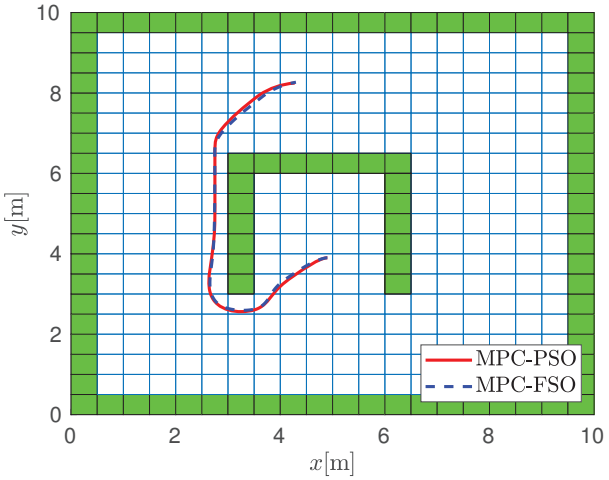

Fig. 9. Comparison of model predictive control with PSO and FSO on interpolated potential function.

velocity profile is obtained by Particle Swarm Optimization, while Fix Set Optimization results in oscillating velocities and slower motion. However, both methods produce similar final trajectories from start to goal since optimization is guided by the same objective function.

IV. Conclusion

The article presents wheeled mobile robot navigation approach that applies potential field as a navigation function. This navigation function is obtained from discrete graph-based search algorithm and can also adapt to the environment changes on-line. To obtain continuous navigation function a simplex based interpolation is done. The resulting navigation function features single minimum in the goal state and no other local minima. It is applied first in the simple gradient-direction following controller which can result in a slightly oscillating trajectory because it considers the gradient of navigation function in current location only. This is improved by the use of the MPC where optimal controls are determined using PSO optimization and by a fix set optimization.

Future investigation will consider other methods for navigation function interpolation such as bi-cubic interpolation that would make its surface smother. Current approach can easily be used online with sampling frequency 10 Hz or more. Furthermore, conditions for convergence guarantee will be considered.

References

[1] J.-C. Latombe, *Robot Motion Planning*. Kluwer Academic Publishers, 1991.

[2] H. Choset, K. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki, and S. Thrun, *Principles of robot motion: theory, algorithms, and implementations*, R. C. Arkin, Ed. MIT Press, 2005.

[3] T. T. Mac, C. Copot, D. T. Tran, and R. D. Keyser, "Heuristic approaches in robot path planning: A survey," *Robotics and Autonomous Systems*, vol. 86, pp. 13 – 28, 2016.

[4] P. Ogren and N. E. Leonard, "A convergent dynamic window approach to obstacle avoidance," *IEEE Transactions on Robotics*, vol. 21, no. 2, pp. 188–195, April 2005.

[5] M. Seder, M. Baotić, and I. Petrović, "Receding horizon control for convergent navigation of a differential drive mobile robot," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 2, pp. 653–660, March 2017.

[6] G. Klančar and I. Škrjanc, "Tracking-error model-based predictive control for mobile robots in real time," *Robotics and Autonomous Systems*, vol. 55, no. 6, pp. 460 – 469, 2007.

[7] R. K. Mandava, S. Bondada, and P. R. Vundavilli, "An optimized path planning for the mobile robot using potential field method and pso algorithm," in *Soft Computing for Problem Solving*, J. C. Bansal, K. N. Das, A. Nagar, K. Deep, and A. K. Ojha, Eds. Singapore: Springer Singapore, 2019, pp. 139–150.

[8] H. Lu and Y. Yin, "Fast path planning for autonomous ships in restricted waters," *Applied Sciences*, vol. 8, 12 2018.

[9] F. A. Cosio and M. P. Castaneda, "Autonomous robot navigation using adaptive potential fields," *Mathematical and Computer Modelling*, vol. 40, no. 9, pp. 1141 – 1156, 2004.

[10] M. Guerra, D. Efimov, G. Zheng, and W. Perruquetti, "Avoiding local minima in the potential field method using input-to-state stability," *Control Engineering Practice*, vol. Volume 55, pp. 174–184, 07 2016.

[11] M. G. Park and M. C. Lee, "A new technique to escape local minimum in artificial potential field based path planning," *KSME International Journal*, vol. 17, no. 12, pp. 1876–1885, Dec 2003.

[12] G. Klančar and G. Mušič, "Fast marching method based path planning for wheeled mobile robots," in *Conference proceedings, 8th International Mediterranean Latin American Modelling Multiconference*, Rome, Italy, 2011, pp. 118–126.

[13] H. Chen and W. Yan, "Flux characteristics analysis of a double-sided switched reluctance linear machine under the asymmetric air gap," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 12, pp. 9843–9852, Dec 2018.