Particle Swarm Optimization of Potential Fields for Obstacle Avoidance

Rainer Palm
Örebro University
SE-70182 Örebro, Sweden
rub.palm@t-online.de

Abdelbaki Bouguerra
Örebro University
SE-70182 Örebro, Sweden
abdelbaki.bouguerra@oru.se

***Abstract*—This paper addresses the safe navigation of multiple nonholonomic mobile robots in shared areas. Obstacle avoidance for mobile robots is performed by artificial potential fields and special traffic rules. In addition, the behavior of mobile robots is optimized by particle swarm optimization (PSO). The control of non-holonomic vehicles is performed using the virtual leader principle together with a local linear controller.**

***Index Terms*—Mobile robots, obstacle avoidance, potential field, particle swarm optimization**

I. INTRODUCTION

In the last two decades several methods of robot navigation and obstacle avoidance have been discussed. One of the most prominent methods for obstacle avoidance is the artificial potential field method (see [1]). Borenstein and Koren present a review on this method addressing its advantages and disadvantages with respect to stability and deadlocks (see [2]). Another approach can be found in ([3]) where local groups of robots share information on common potential field regions for navigation among static and dynamic obstacles. Further research results regarding navigation of non-holonomic mobile robots can be found in [4] and [5]. The execution of robot tasks based on semantic domain-knowledge has been reported in detail by [6].

These examples show the wide variety of methods dealing with different subtasks like

- go to target
- avoid obstacle
- follow traffic rules

Achieving different tasks at the same time requires a decentralized optimization leading necessarily to a different weighting of the tasks. Multi-agent control as a decentralized approach can handle the optimization of tasks for a large number of complex local systems more efficiently than centralized approaches.

In the context of mobile robot navigation, combinations of competing tasks, that should be optimized, can be manifold, for example the presence of traffic rules and the necessity for avoiding an obstacle by using artificial potential fields at the same time. Another case is the accidental meeting of more than two robots within a small area. This requires a certain minimum distance between the robots and appropriate (smooth) maneuvers to keep stability of trajectories to be tracked. These situations require additional approaches to enhance classical methods like artificial potential fields.

The current paper addresses just this point where optimization takes place between ”competing” potential fields of mobile robots: Based on appropriate optimization methods some potential fields are strengthened, some are weakened depending on the local situation.

One of these methods is Particle Swarm Optimization (PSO) first published by Kennedy and Eberhart [7]. PSO is an evolutionary algorithm that imitates social behavior of bird flocking or fish schooling. Raja et al gave a review on optimal path planning ot mobile robots where PSO played a prominent role under the methods considered [8]. Gong et al described a multi-objective PSO for path planning [9]. Min et al proposed a mathematical model using PSO and the so-called collision cone approach for obstacle avoidance [10].

Another promising approach to cope with large decentralized systems is the market-based optimization (MBO). MBO imitates economical systems where producer and consumer agents both compete and cooperate on a market of commodities. A combination of artificial potential fields and MBO has already been proposed by Palm et al [11], [12]. The main topic of this paper is the combination of the artificial potential field method with PSO and the design of a low-level controller for the non-holonomic vehicle using the virtual leader principle.

The paper is organized as follows. Section II deals with navigation principles applied to the robot task. In section III the modeling and control of a non-holonomic vehicle is presented, and the navigation and obstacle avoidance using potential fields in the framework of a multi-robot system is outlined. In section IV the Particle Swarm Optimization approach (PSO) is presented and the connection between the both PSO approach and the system to be controlled is outlined. Section V shows simulation results, and Section VI draws conclusions and highlights future work.

II. NAVIGATION PRINCIPLES

Navigation principles for a mobile robot (platform) P_i are *heuristic rules* to perform a specific task under certain restrictions regarding the environment, obstacles O_j, and other robots P_j. Each platform P_i has an estimation about position/orientation of itself and the target T_i. The position of another platform P_j relative to P_i can be measured if it lies within the sensor cone of P_i. Let, for example, mobile robots (platforms) P_1, P_2, and P_3 move from their starting points to targets T_1, T_2, and T_3, respectively, whereas collisions should be avoided. Four navigation principles are used here

1. Move to target T_i
2. Avoid an obstacle O_j (static or dynamic) if it appears in the sensor cone at a certain distance.
3. Decrease speed if a dynamic obstacle O_j (platform) comes from the right
4. Move to the right if the obstacle angles β (see [13]) of two approaching platforms are small (e.g. $\beta < 10°$).

Except the *heading-to-target movement* all other navigation calculations and actions take place in the local coordinate system of platform P_i. The positions of obstacles (static or dynamic) O_j or of other platforms P_j are also formulated in the local frame of platform P_i.

III. Navigation and obstacle avoidance using potential fields

A. Modeling of the system

We consider a non-holonomic rear-wheel driven vehicle with the kinematics of a car. The kinematic of the non-holonomic vehicle (see Fig.1) is described by

$$\begin{aligned}\dot{q}_i &= R_i(q_i)\cdot u_i \\ q_i &= (x_i, y_i, \Theta_i, \phi_i)^T \\ R_i(q_i) &= \begin{pmatrix} \cos\Theta_i & 0 \\ \sin\Theta_i & 0 \\ \frac{1}{l_i}\cdot\tan\phi_i & 0 \\ 0 & 1 \end{pmatrix}\end{aligned} \tag{1}$$

where
$q_i \in \Re^4$ - state vector
$u_i = (u_{1i}, u_{2i})^T \in \Re^2$ - control vector, pushing/steering speed
$x_{ip} = (x_i, y_i)^T \in \Re^2$ - position vector of platform P_i
Θ_i - orientation angle
ϕ_i - steering angle
l_i - length of vehicle

1) Virtual leader: In most of the control methods the target is located at a far distance from the vehicle to be controlled. In contrast to this we introduce a 'virtual' vehicle that moves in front of the 'real' one (see also [14]). The virtual vehicle (platform) acts as trajectory generator that generates the position for the real platform at every time step on the basis of starting and target (end) position, obstacles to be avoided, other platforms to be taken into account etc. In this context, the virtual leader can also be considered as an ideal trajectory follower. The dynamics of the virtual platform is designed as a first order system that automatically avoids abrupt changes in position and orientation

$$\dot{v}_{vi} = k_{vi}(v_{vi} - v_{di}) \tag{2}$$

$v_{vi} \in \Re^2$ - velocity of virtual platform P_i
$v_{di} \in \Re^2$ - desired velocity of virtual platform P_i
$k_{vi} \in \Re^{2\times2}$ - damping matrix (diagonal)

v_{di} is composed of the tracking velocity v_{ti} and velocity terms due to artificial potential fields from obstacles and other platforms (23). The tracking velocity is designed as a control term

$$v_{ti} = k_{ti}(x_{ip} - x_{ti}) \tag{3}$$

$x_{ti} \in \Re^2$ - position vector of target T_i
$x_{ip} \in \Re^2$ - position vector of platform P_i
$k_{ti} \in \Re^{2\times2}$ - gain matrix (diagonal)

B. Design of a low level control law - Kinematical solution

In the following the motion of the virtual leader is assumed as a slowly time varying process compared to the time constant of the nonholonomic vehicle which leads to a so-called time-frozen situation. Therefore the introduction of a virtual platform enables us to design a linear control law in a leader-follower scenario. In this context a new variable γ_i' is introduced in the follower's frame (see Fig. 1)

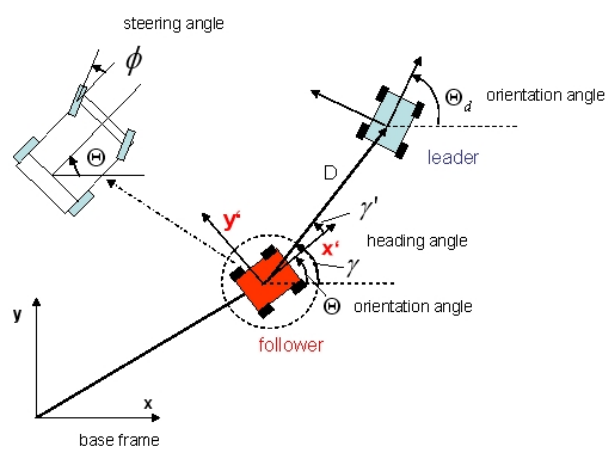

Fig. 1. Leader follower principle

$$\gamma_i' = arctan\frac{y_{di}'}{x_{di}'} \tag{4}$$

Differentiation of (4) yields

$$\dot{\gamma}_i' = \frac{cos^2\gamma_i'}{x_{di}'}(\dot{y}_{di}' - \dot{x}_{di}' \cdot tan\gamma_i') \tag{5}$$

With

$$\begin{aligned}\dot{x}_{di}' &= v_{Di}\cdot cos(\Theta_{di} - \Theta_i) \\ \dot{y}_{di}' &= v_{Di}\cdot sin(\Theta_{di} - \Theta_i)\end{aligned}$$

where v_{Di} is the leader's velocity in the follower's frame we finally obtain

$$\begin{aligned}\dot{\gamma}_i' &= \frac{sin(\Theta_{di} - \gamma_i)}{D_i}\cdot v_{Di} \\ \gamma_i &= \gamma_i' - \Theta_i\end{aligned} \tag{6}$$

$D_i = \sqrt{(x_i - x_{di})^2 + ((y_i - y_{di})^2}$ is the distance between leader and follower.
Θ_{di} is the orientation angle of the leader.

From (6), (5) and (1) we then obtain

$$\dot{\gamma}_i = \frac{sin(\Theta_{di} - \gamma_i)}{D_i} \cdot v_{Di} + v_{di} \cdot \frac{tan\phi_i}{l_i} \qquad (7)$$

In the following *pushing force control* and *steering force control* to be applied to the follower are addressed separately.

1) Pushing force control: The pushing force is composed by the sum of the desired velocity of the leader v_{di} and an additional integral term

$$u_{1i} = v_{di} + k_{Ii} \cdot \int_t (D_i - D_{di})dt \qquad (8)$$

D_{di} - desired distance between leader and follower

2) Steering force control: The following linear steering control law is a composition of

- orientation control
- steering angle control
- heading angle control

$$\begin{aligned} u_{2i} = \dot{\phi}_i &= k_{gain1_i}(\Theta_{di} - \Theta_i) + k_{gain2_i}(\phi_{di} - \phi_i) \\ &\quad -k_{gain3_i}(\Theta_i - \gamma_i) \end{aligned} \qquad (9)$$

where $\phi_{di} = \Theta_{di} - \Theta_i$ is the desired steering angle. Rewriting (9) into

$$\begin{aligned} \dot{\phi}_i &= -C_i \cdot \phi_i + K1_i \cdot (\Theta_i - \Theta_{di}) + K2_i \cdot (\Theta_i - \gamma_i) \\ C_i &= k_{gain2_i} \\ K1_i &= -(k_{gain1_i} + k_{gain2_i}) \\ K2_i &= -k_{gain3_i} \end{aligned} \qquad (10)$$

Summarizing the steering equations (1), (7), and (9) we get

$$\begin{aligned} \dot{\Theta}_i &= \frac{v_{di}}{l_i} \cdot \tan\phi_i \\ \dot{\phi}_i &= -C_i \cdot \phi_i + K1_i \cdot (\Theta_i - \Theta_{di}) + K2_i \cdot (\Theta_i - \gamma_i) \\ \dot{\gamma}_i &= \frac{sin(\Theta_{di} - \gamma_i)}{D_i} \cdot v_{Di} + v_{di} \cdot \frac{tan\phi_i}{l_i} \end{aligned} \qquad (11)$$

Suppose that at the begin of the motion leader and follower are close each other. Then equations (11) can be linearized

$$\begin{aligned} \dot{\Theta}_i &= \frac{v_{di}}{l_i} \cdot \phi_i \\ \dot{\phi}_i &= -C_i \cdot \phi_i + (K1_i + K2_i) \cdot \Theta_i - K2_i \cdot \gamma_i - K1_i \cdot \Theta_{di} \\ \dot{\gamma}_i &= \frac{v_{di}}{l_i} \cdot \phi_i - \frac{v_{Di}}{D_i} \cdot \gamma_i + \frac{v_{Di}}{D_i} \cdot \Theta_{di} \end{aligned} \qquad (12)$$

where $tan\phi_i \approx \phi_i$ and $sin(\Theta_i - \gamma_i) \approx \Theta_i - \gamma_i$.

Equation (12) can be written in the compact form

$$\begin{aligned} \dot{\tilde{q}}_i &= A_i \tilde{q}_i + B_i \tilde{u}_i \\ \tilde{q}_i &= (\Theta_i, \phi_i, \gamma_i)^T \\ \tilde{u}_i &= \Theta_{di} \end{aligned} \qquad (13)$$

where

$$\begin{aligned} A_i &= \begin{pmatrix} 0 & \frac{v_{di}}{l_i} & 0 \\ (K1_i + K2_i) & -C_i & -K2_i \\ 0 & \frac{v_{di}}{l_i} & -\frac{v_{Di}}{D_i} \end{pmatrix} \\ B_i &= \begin{pmatrix} 0 \\ -K1_i \\ \frac{v_{Di}}{D_i} \end{pmatrix} \end{aligned} \qquad (14)$$

Equation (13) is stable if A_i is *Hurwitz*. The gains $K1_i, K2_i$ and, with this, $k_{gain1}, k_{gain2}, k_{gain3}$ are designed by pole placement. From the determinant of matrix $A_i - \lambda E$, we obtain

$$\begin{aligned} |A_i - \lambda E| &= \begin{vmatrix} 0 - \lambda & \frac{v_{di}}{l_i} & 0 \\ (K1_i + K2_i) & -(C_i + \lambda) & -K2_i \\ 0 & \frac{v_{di}}{l_i} & -(\frac{v_{Di}}{D_i} + \lambda) \end{vmatrix} \\ &= 0 \end{aligned} \qquad (15)$$

where $E \in \Re^{3\times3}$ - identity matrix. Solving equation (15) we calculate the relation between the three poles $\lambda_1, \lambda_2, \lambda_3$, the system parameters and the control parameters to be designed

$$\begin{aligned} \lambda_1\lambda_2\lambda_3 &= \frac{v_{di}}{l_i} \cdot \frac{v_{Di}}{D_i} \cdot (K1_i + K2_i) \\ \lambda_1\lambda_2 + \lambda_1\lambda_3 + \lambda_2\lambda_3 &= \frac{v_{Di}}{D_i} \cdot C_i - \frac{v_{di}}{l_i} \cdot K1_i \\ \lambda_1 + \lambda_2 + \lambda_3 &= -(C_i + \frac{v_{Di}}{D_i}) \end{aligned} \qquad (16)$$

from which we obtain

$$\begin{aligned} C_i &= -\frac{v_{Di}}{D_i} - (\lambda_1 + \lambda_2 + \lambda_3) \\ K1_i &= \frac{l_i}{v_{di}} \cdot (\frac{v_{Di}}{D_i} C_i - (\lambda_1\lambda_2 + \lambda_1\lambda_3 + \lambda_2\lambda_3)) \quad (17) \\ K2_i &= \frac{l_i}{v_{di}} \cdot \frac{D_i}{v_{Di}} \cdot \lambda_1\lambda_2\lambda_3 - K1_i \end{aligned} \qquad (18)$$

The original gains $k_{gain1_i}, k_{gain2_i}, k_{gain3_i}$ can be obtained using (10)

$$K1_i + K2_i = -(k_{gain1_i} + k_{gain2_i} + k_{gain3_i}) \qquad (19)$$

Let further $K_{tot i} = -k_{gain2_i} - (K1_i + K2_i)$ and

$$\begin{aligned} k_{gain1_i} &= \alpha \cdot K_{tot i} \\ k_{gain3_i} &= (1 - \alpha) \cdot K_{tot i} \end{aligned} \qquad (20)$$

with the free design parameter $\alpha \in [0, 1]$, we obtain with (10) and (20) the three gains $k_{gain1_i}, k_{gain2_i}, k_{gain3_i}$ that guarantee a stable motion of the follower along the trajectory of the leader. The excellent tracking quality of the proposed control can be observed by the simulation examples especially for 'moving on lines' (see Section V Figs. 2 - 5).

C. Introduction of artificial potential fields

Speaking in the following of 'forces' does not mean forces in the physical sense but 'virtual forces' or 'artificial forces'. Repulsive forces exist between platform P_i and obstacle O_j leading to repulsive velocities

$$v_{ij_{ob}} = -c_{ij_{ob}}(x_{ip} - x_{j_{ob}})d_{ij_{ob}}^{-2} \quad (21)$$

$v_{ij_{ob}} \in \Re^2$ - repulsive velocity vector between platform P_i (leader) and obstacle O_j
$x_{j_{ob}} \in \Re^2$ - position vector of obstacle O_j
$d_{ij_{ob}} \in \Re$ - Euclidian distance between platform P_i and obstacle O_j
$c_{ij_{ob}} \in \Re^{2\times 2}$ - gain matrix (diagonal)

Repulsive forces also appear between platforms P_i and P_j from which we get the repulsive velocities

$$v_{ij_p} = -c_{ij_p}(x_{ip} - x_{j_p})d_{ij_p}^{-2} \quad (22)$$

$v_{ij_p} \in \Re^2$ - repulsive velocity between platforms P_i and P_j
$d_{ij_p} \in \Re$ - Euclidian distance between platforms P_i and P_j
$c_{ij_p} \in \Re^{2\times 2}$ - gain matrix (diagonal)

The resulting velocity v_{di} is

$$v_{di} = v_{ti} + \sum_{j=1}^{m_{ob}} v_{ij_{ob}} + \sum_{j=1}^{m_p} v_{ij_p} \quad (23)$$

where m_{ob}, and m_p are the numbers of contributing obstacles, and platforms, respectively. In general, artificial force fields are switched on/off according to the actual scenario: distance between interacting systems, state of activation according to the sensor cones of the platforms, positions and velocities of platforms w.r.t. to targets, obstacles and other platforms. All calculations of the velocity components (1)-(23), angles and sensor cones are formulated in the local coordinate systems of the platforms.

IV. Optimization of the behavior of the mobile robot system

The behavior of the multiple mobile robot system is optimized by an appropriate weighting of the repulsive forces/velocities v_{ij_p} between the platforms. The desired motion of platform P_i is then described by

$$v_{d_i} = vo_i + \sum_{j=1, i\neq j}^{m_p} w_{ij} v_{ij_p} \quad (24)$$

w_{ij} are weighting factors for repelling forces between platforms P_i and P_j. vo_i is a combination of

- tracking velocity depending on distance between platforms P_i and targets T_i
- repulsive and control terms between platforms P_i and obstacles O_j
- Traffic rules

The goal is to change the weights w_{ij} to generate a smooth dynamical behavior in a common working area. This can be achieved by minimizing of some energy function representing the interaction of platforms and obstacles. One possible option for tuning the weights w_{ij} is to find a global optimum over all contributing platforms. This, however, is rather difficult especially in the case of many interacting platforms. Instead, the multi-agent approach has been preferred. In the following the so-called Particle Swarm Optimization (PSO) is presented leading to the optimization of the robot's behavior when acting in a common working area. It has to be emphasized that the whole optimization process is done online.

A. Particle swarm optimization (PSO)

PSO is an optimization method that simulates the behavior of bird flocking generated by a swarm of birds searching for food. In PSO each single bird in the swarm is simulated by an agent - a so-called particle. Each particle has a fitness value that is evaluated by a fitness function to be optimized. For the optimization of the behavior of mobile robots PSO uses

1. a cost function J_i to be optimized like

$$J_i = \sum_{j=1}^{m_p} J_{ij} \rightarrow optimum \quad (25)$$
$$J_{ij} = a_{ij} + b_{ij}w_{ij} + c_{ij}w_{ij}^2$$

   a_{ij}, b_{ij}, c_{ij} - system parameters
   w_{ij} - variable to be optimized
   m_p - number of robots in a shared area

2. a group of particles (agents) Pt_i forming a so-called swarm.

Each particle (agent) Pt_i varies w_{ij} randomly within a given interval while calculating J_{ij} and checking J_i regarding its optimum (minimum or maximum). Values of w_{ij} resulting in a J_i closer to the optimum as before are selected as "better" ones. Do this for each robot $i = 1...m_p$ in the shared area. This procedure is - so to speak - a competition between particles (agents) Pt_m during which the particle Pt_m and the associated w_{ij_m} with the best $J_{iopt_{tot}}$ wins the game.

The random process works as follows:

a) Calculate initial $J_{ij_{init_m}}$'s and an initial J_{iinit_m} for each particle Pt_m and pick the best J_{iopt_m} for each particle Pt_m and a corresponding $p_m = w_{ij_{opt_m}}$. Then pick the best $J_{iopt_{tot}}$ among all J_{iopt_m} and a corresponding $q_m = w_{ij_{opt_{tot}}}$.
b) Calculate a "gradient" $\nu(k)$ with a help of which the variable $w_{ij}(k)$ moves towards the optimum after each optimization step k.

The updating formula for $\nu(k)$ reads

$$\begin{aligned}\nu(k+1) = \omega\cdot\nu(k) + c_1\cdot r_1\cdot(p_m(k) - w_{ij}(k)) \\ + c_2\cdot r_2\cdot(q_m(k) - w_{ij}(k))\end{aligned} \quad (26)$$

and for $w_{ij}(k)$

$$w_{ij}(k+1) = w_{ij}(k) + \nu(k+1) \quad (27)$$

$\nu(k)$ - gradient
k - optimization step

$r_1, r_2 \in [0,1]$ - random variables
$\omega, c_1, c_2 \geq 0$ - free parameters
$\omega \cdot \nu(k)$ - inertia weight
$c_1 \cdot r_1 \cdot (p_m(k) - w_{ij}(k))$ - personal weight (local)
$c_2 \cdot r_2 \cdot (q_m(k) - w_{ij}(k))$ - social weight (global)

The *inertia weight* plays the role of damping of the learning process. The *personal weight* gives more/less weight to the particle than to the swarm whereas the *social weight* gives more/less weight to the swarm. The whole process is iterative: after either a given number of iterations or falling below some given threshold the optimization process is finished.

From the system equation (24) we define further a local energy function to be minimized

$$\begin{aligned} \tilde{J}_{ij} &= v_{d_i}^T v_{d_i} \\ &= a_{ij} + b_{ij} w_{ij} + c_{ij} (w_{ij})^2 \rightarrow min \end{aligned} \quad (28)$$

where $\tilde{J}_{ij} \geq 0, a_{ij}, c_{ij} > 0$.

B. Determination of the system parameters a_{ij}, b_{ij} and c_{ij}

The calculation of the system parameters a_{ij}, b_{ij}, and c_{ij} is based on the equation of the system of mobile robots (24)

$$v_{d_i} = vo_i + \sum_{j=1, i \neq j}^{m_p} w_{ij} v_{ij_p} \quad (29)$$

where vo_i is a subset of the RHS of (23) - a combination of different terms (tracking velocity, repelling and rotational forces between platforms and obstacles, traffic rules etc.), v_{ij_p} reflects the repelling forces between platforms P_i and P_j.

The *local* energy funcion reflects only the energy of a pair of two interacting platforms P_i and P_j

$$\begin{aligned} \tilde{J}_{ij} &= vo_i^T vo_i + (\sum_{k=1, k \neq i,j}^{m_p} w_{ik} v_{ik_p})^T (\sum_{k=1, k \neq i,j}^{m_p} w_{ik} v_{ik_p}) \\ &+ 2 \sum_{k=1, k \neq i,j}^{m_p} w_{ik} vo_i^T v_{ik_p} \\ &+ 2 w_{ij} (vo_i^T + \sum_{k=1, k \neq i,j}^{m_p} w_{ik} v_{ik_p}^T) v_{ij_p} \\ &+ w_{ij}^2 (v_{ij_p}^T v_{ij_p}) \end{aligned} \quad (30)$$

Comparison of (30) and (28) yields

$$\begin{aligned} a_{ij} &= vo_i^T vo_i + (\sum_{k=1, k \neq i,j}^{m} w_{ik} v_{ik_p})^T (\sum_{k=1, k \neq i,j}^{m} w_{ik} v_{ik_p}) \\ &+ 2 \sum_{k=1, k \neq i,j}^{m} w_{ik} vo_i^T v_{ik_p} \\ b_{ij} &= 2 (vo_i^T + \sum_{k=1, k \neq i,j}^{m_p} w_{ik} v_{ik_p}^T) v_{ij_p} \\ c_{ij} &= (v_{ij_p}^T v_{ij_p}) \end{aligned} \quad (31)$$

With this the parameters for the computation of the weights w_{ij} and the optimization of J_{ij} over all contributing platforms P_i is given.

V. Simulation results

In our simulations the platforms are supposed to move to static targets while avoiding other platforms and static obstacles at the same time. In the optimization process the number of particles was 10. It could also be observed that an iteration number of 10 led to a sufficiently fast convergence result. The time step of the simulation is $T = 0.05s$. The units ate the axes are m. To determine the quality of the particular approach especially for the short time period of robot interaction in a shared area two performance measures are considered: i) *bending energy*, ii) *smoothness* [15]. Other parameters like traveled distance and completion time were left out. The *bending energy* is measured by $bend_i = \sum_k curv_i(k)^2$ where $curv_i(k)$ is the curvature of the ith trajectory at time step k. "The *bending energy* can be understood as the energy needed to bend a rod to the desired shape" [15]. The *smoothness* is measured by the sum of absolute values of the change in curvatures $d(curv_i) = \sum_k |curv_i(k+1) - curv_i(k)|$. Measurements and calculations are performed at each point of the trajectories and numerically integrated along them. Figures 2 - 5 show the trajectories of the platforms. To show how optimization works in difficult situations the targets are supposed to change their positions drastically after a certain number of steps. The following formula serves as an intuitive performance measure:

$$perf_i = bend_i \cdot d(curv_i) \quad i = 1...3 \quad (32)$$

The smaller/larger $perf_i$ is, the better/worse is the performance of the tracking example. Finally, to decide about the performance of the total experiment the sum over all 3 trajectories is formed

$$perf_{tot} = \sum_i perf_i \quad i = 1...3 \quad (33)$$

Other results with targets moving on circles are shown in Figs. 6 and 7. The result for the entire experiment is shown in Table I from which we conclude that PSO improves the total performance significantly.

TABLE I
SIMULATION RESULTS

experiment	no PSO	with PSO
lines	$perf_{tot,noopt} = 1555.81$	$perf_{tot,PSO} = 57.29$
circles	$perf_{tot,noopt} = 27.6 \cdot 10^{10}$	$perf_{tot,PSO} = 7.93 \cdot 10^4$

VI. Conclusions

Navigation and obstacle avoidance of mobile robots are performed by artificial potential fields and traffic rules. The control of the vehicles takes place by using the virtual leader principle and a local linear control strategy. In addition the behavior of mobile robots is optimized by particle swarm optimization (PSO). Optimization takes place when more than two mobile robots act in a common workspace. PSO is an optimization method that simulates the behavior of bird flocking or fish schooling. By means of weighting factors - optimized by PSO - potential fields are strengthened or weakened depending on

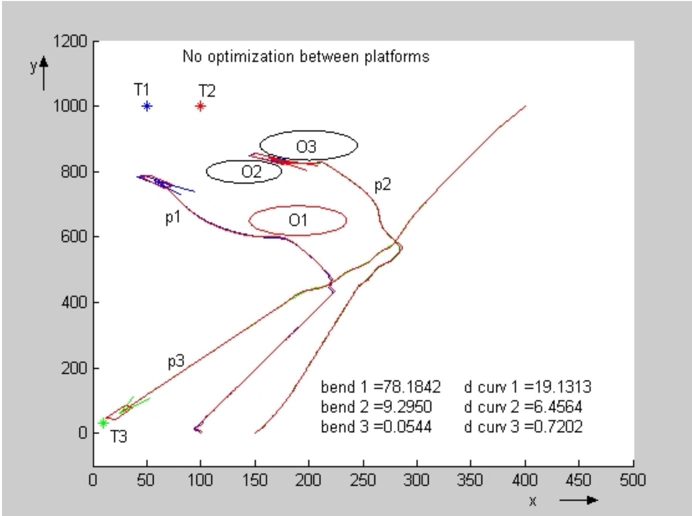

Fig. 2. Moving on lines with change of targets, no optimization

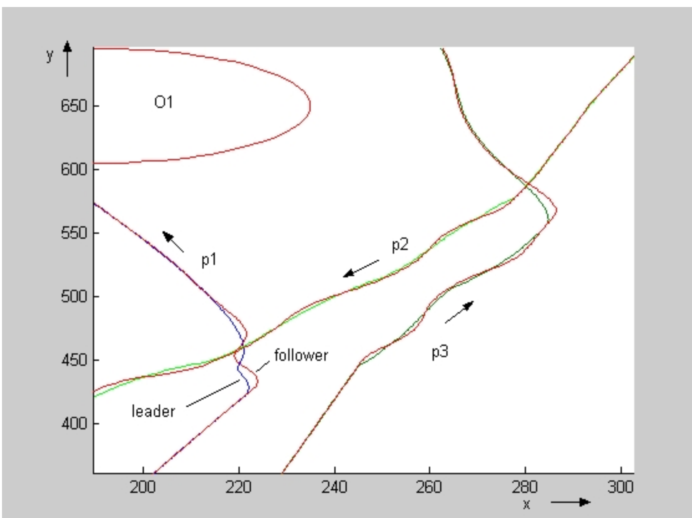

Fig. 3. Zoom in for moving on lines, no optimization

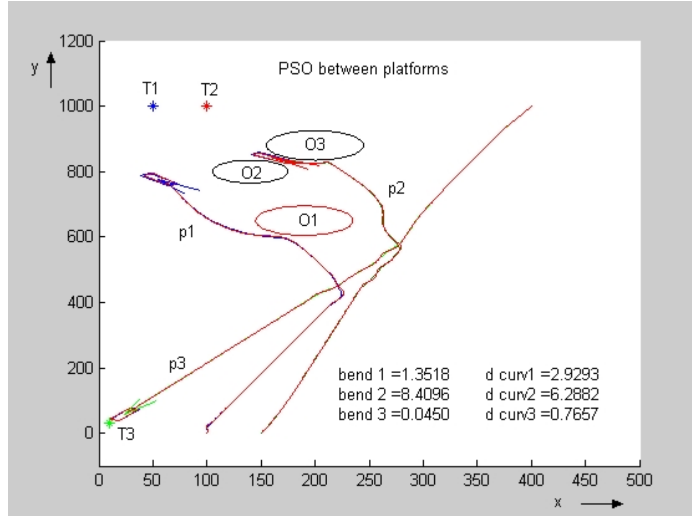

Fig. 4. Moving on lines with change of targets, with PSO

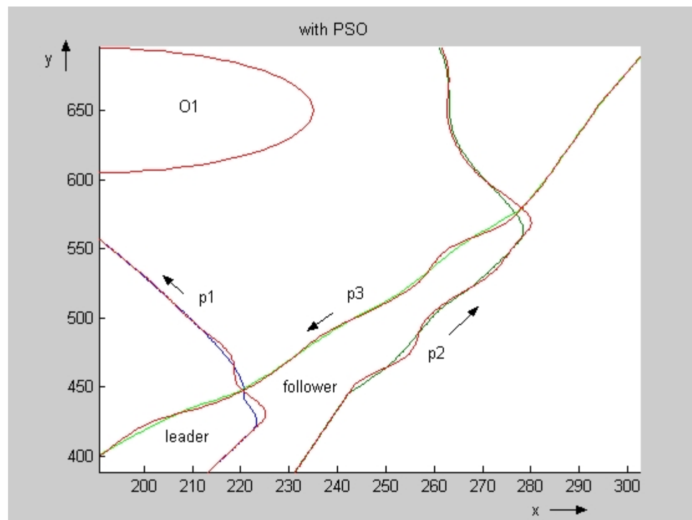

Fig. 5. Zoom in for moving on lines, with PSO

the actual scenario allowing smooth motions in such situations. Simulation experiments with simplified robot kinematics and dynamics have shown the feasibility of the presented method. A future aspect of this work is a more realistic simulation followed by an implementation of the algorithm on a set of real mobile robots.

REFERENCES

[1] O. Khatib. Real-time 0bstacle avoidance for manipulators and mobile robots. *IEEE Int. Conf. On Robotics and Automation,St. Loius,Missouri, 1985*, page 500505, 1985.

[2] Y. Koren and J. Borenstein. Potential field methods and their inherent limitations for mobile robot navigation. *Proceedings of the IEEE Conference on Robotics and Automation, Sacramento, California*, pages 1398–1404, April 7-12, 1991.

[3] J.L. Baxter, E.K. Burke, J.M. Garibaldi, and M. Norman. Multi-robot search and rescue: A potential field based approach. *Autonomous Robots and Agents Series: Studies in Computational Intelligence , Vol. 76, Mukhopadhyay, Subhas; Sen Gupta, Gourab (Eds.), Springer-Verlag*, pages 9–16, 2007.

[4] M. J.Sorensen. Artificial potential field approach to path tracking for a nonholonomic mobile robot. *Proceedings of the 11th Mediteranean Conference on Control And Automation. Rhodes, Greece, 2003*, 2003.

[5] J.Alonso-Mora, A. Breitenmoser, M.Rufli, P. Beardsley, and R. Siegwart. Optimal reciprocal collision avoidance for multiple non-holonomic robots. *Proc. of the 10th Intern. Symp. on Distributed Autonomous Robotic Systems (DARS), Switzerland*, Nov 2010.

[6] A. Bouguerra. Robust execution of robot task-plans: A knowledge-based approach. *Ph.D. Thesis,Oerebro University*, 2008.

[7] J.Kennedy and R.C. Eberhart. Particle swarm optimization. *Proc. of the IEEE Intern. Conf. on Neural Networks, Perth, Australia*, pages 1942–1948, 1995.

[8] P.Raja and S.Pugazhenthi. Optimal path planning of mobile robots: A review. *Intern. Journal of Physical Sciences*, 7(9):1314–1320, Feb 2012.

[9] D.W. Gong, J.H.Zhang, and Y.Zhang. Multi-objective particle swarm optimization for robot path planning in environment with danger sources. *J. Comput.*, 6(8):1554–1561, 2011.

[10] H.Q. Min, J.H. Zhu, and X.J. Zheng. Obstacle ovoidance with multi-objective optimization by pso in dynamic environment. *Proc. of the IEEE Intern. Conf. on Machine Learning and Cybernetics, Guangzhou, PR China*, pages 2950–2956, 2005.

[11] R. Palm. Synchronization of decentralized multiple-model systems by

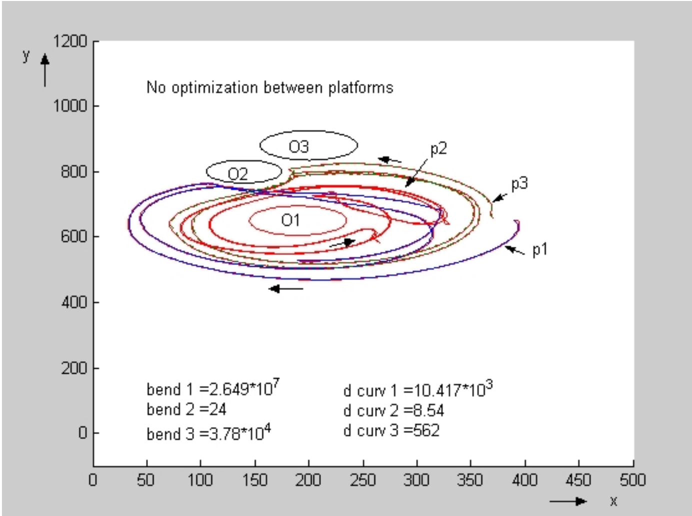

Fig. 6. moving on circles, no optimization

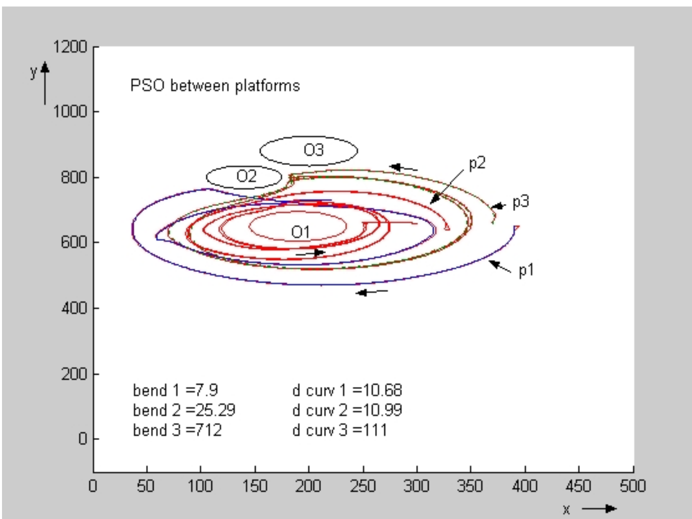

Fig. 7. moving on circles, with PSO

market-based optimization. *IEEE Trans Syst Man Cybern B, Vol. 34*, pages 665–72, Feb 2004.

[12] R.Palm and A. Bouguerra. Navigation of mobile robots by potential field methods and market-based optimization. *ECMR 2011*, Oerebro, Sweden., Sept. 7-9, 2011.

[13] B. R. Fajen and W. H. Warren. Behavioral dynamics of steering, obstacle avoidance, and route selection. *Journal of Experimental Psychology: Copyright by the American Psychological Association, Inc. Human Perception and Performance, Vol. 29, No. 2*, page 343362, 2003.

[14] N. E. Leonard and E. Fiorelli. Virtual leaders, artificial potentials and coordinated control of groups. *Proc. of the 40th IEEE Conf. on Decision and Control, Orlanda, Florida USA*, pages 2968–2973, Dec 2001.

[15] N. Munoz, J.Valencia, and N. Londono. Evaluation of navigation of an autonomous mobile robot. *Proc. of int. workshop on performance metrics for intelligent systems workshop (PerMIS)*, pages 15–21, 2007.