Modeling and Motion Planning for a Population of Mobile Robots

Qirong Tang and Peter Eberhard

Institute of Engineering and Computational Mechanics,
University of Stuttgart, Stuttgart, Germany
[tang, eberhard]@itm.uni-stuttgart.de

Abstract This paper investigates modeling and motion planing for a population of mobile robots. A mechanical model is built for the robot motion planning which is inspired by particle swarm optimization and combined with multibody system dynamics. It uses the augmented Lagrangian multiplier method to treat the constraints and an independent module to handle obstacle avoidance. Simulations show that the robots moving in the environment display the desired behavior well, and so this model and related algorithms will next be transfered to a large scale mobile robotic system.

1 Introduction

Nowadays, controlling of a multi-robot or so called swarm robots is still a challenge in the robotics area despite of its fast development. Thus, several researchers worked on finding methods of modeling and motion planning for such swarm robotic systems.

In recent years, the comparatively new stochastic particle swarm optimization (PSO) algorithms have been applied to many engineering areas, among them also the robotics area. (Doctor et al., 2004) discussed using PSO for multi-robot searching. Their focus was on optimizing the parameters and they did not consider the scalability of the standard PSO for a large numbers of robots. (Hereford and Siebold, 2008) developed a distributed PSO but with shortcomings of non-consideration of obstacles or restriction to just static simple obstacles in the environment and the real robots could only rotate at a specified angle range. (Pugh et al., 2006) contributed a simple PSO version in multi-robot searching and they mainly focused on how to model the biological algorithm.

In contrast to the mentioned publications this paper builds a model and does the motion planning for a population of mobile robots based on PSO combined with multibody systems. It is expected that after some adaption, such models and methods can be adjusted for real mobile robots.

2 Modeling for a Population of Mobile Robots

A population of robots is not simply a group of individuals. The behavior shown by a population of robots should be collective, coordinated and requires information exchange. So far, many algorithms and methods are used in this area, among them traditional and recursive biology inspired methods, see (Tang and Eberhard, 2009). The PSO algorithm is very appealing due to its clear ideas, simple iteration equations, and the possibility to be mapped onto several robots or even swarm robots.

2.1 Basic Particle Swarm Optimization

The original model of PSO, see (Kennedy and Eberhart, 1995), uses the vectors $\Delta\boldsymbol{x}$ and $\boldsymbol{x}$ to denote the particle's 'velocity' and actual position, respectively. The so called 'velocity' of the i-th particle at the $(k+1)$-th iteration can be described with the equation

$$\Delta\boldsymbol{x}_i^{k+1} = \omega\Delta\boldsymbol{x}_i^k + c_1 r_{i,1}^k(\boldsymbol{x}_{i,self}^{best,k} - \boldsymbol{x}_i^k) + c_2 r_{i,2}^k(\boldsymbol{x}_{swarm}^{best,k} - \boldsymbol{x}_i^k) \tag{1}$$

and the position update is done in the traditional PSO algorithm by

$$\boldsymbol{x}_i^{k+1} = \boldsymbol{x}_i^k + \Delta\boldsymbol{x}_i^{k+1}. \tag{2}$$

If consider in a general way and for all n particles, the basic PSO model can be foumulated by

$$\begin{bmatrix} \boldsymbol{x}^{k+1} \\ \dot{\boldsymbol{x}}^{k+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}^k \\ \omega\dot{\boldsymbol{x}}^k \end{bmatrix} + \begin{bmatrix} \dot{\boldsymbol{x}}^{k+1} \\ c_1\boldsymbol{r}_1^k(\boldsymbol{x}_{self}^{best,k} - \boldsymbol{x}^k) + c_2\boldsymbol{r}_2^k(\hat{\boldsymbol{x}}_{swarm}^{best,k} - \boldsymbol{x}^k) \end{bmatrix}. \tag{3}$$

Detailed interpretation for this structure and related parameters please see (Tang and Eberhard, 2009).

2.2 Build Mechanical Model for Practical Use in a Population of Mobile Robots

This study wants to interpret the PSO algorithm as providing the required forces in the view of multibody system dynamics rather than a mathematical optimization tool as usual since the application purpose is swarm mobile robots motion planning. Each particle (robot) is considered as one body in a multibody system which is influenced by forces and torques from other bodies in the system but without direct mechanical constraints between them. The forces are artificially created by corresponding drive controllers. From another point of view, based on the Newton-Euler equations, the general form of equation of motion for swarm mobile robots can be formulated as

$$\boldsymbol{M}\ddot{\boldsymbol{x}} + \boldsymbol{k} = \boldsymbol{q} \qquad \text{or} \qquad \ddot{\boldsymbol{x}} = \boldsymbol{M}^{-1}(\boldsymbol{q} - \boldsymbol{k}) = \boldsymbol{M}^{-1}\boldsymbol{F}. \tag{4}$$

For a free system without joints, $\boldsymbol{M} = \boldsymbol{diag}(m_1\boldsymbol{I}_3, m_2\boldsymbol{I}_3, \cdots, m_n\boldsymbol{I}_3, \boldsymbol{J}_1, \boldsymbol{J}_2, \cdots, \boldsymbol{J}_n) = \boldsymbol{M}^T \geq 0$ is the inertia matrix collecting the masses and moments of inertias of the particles, $\ddot{\boldsymbol{x}} = [\boldsymbol{a} \;\; \boldsymbol{\alpha}]^T$ is the generalized acceleration, $\boldsymbol{k}$ is a term which comes from the Euler equation, and $\boldsymbol{q}$ contains forces and torques acting on the robots. The force $\boldsymbol{F}$ is determined from three PSO-related parts which are defined as

$$\boldsymbol{f}_1^k = -\boldsymbol{h}_{f_1}^k \left(\boldsymbol{x}^k - \boldsymbol{x}_{self}^{best,k}\right), \;\; \boldsymbol{f}_2^k = -\boldsymbol{h}_{f_2}^k \left(\boldsymbol{x}^k - \hat{\boldsymbol{x}}_{swarm}^{best,k}\right), \;\; \boldsymbol{f}_3^k = -\boldsymbol{h}_{f_3}^k \dot{\boldsymbol{x}}^k, \tag{5}$$

with the matrices

$$\boldsymbol{h}_{f_i}^k = \boldsymbol{diag}(h_{1,f_i}^k\boldsymbol{I}_3, h_{2,f_i}^k\boldsymbol{I}_3, \cdots, h_{n,f_i}^k\boldsymbol{I}_3, \boldsymbol{0}_{3n}), \qquad i = 1,2,3. \tag{6}$$

Here $\boldsymbol{I}_3$ is a 3×3 unit matrix, $\boldsymbol{0}_{3n}$ is a $3n \times 3n$ zero matrix. The forces $\boldsymbol{f}_1^k$ and $\boldsymbol{f}_2^k$ are attraction forces from the last best position of the robot itself and the last swarm best robot position and are proportional to their distances. The vector $\boldsymbol{f}_3^k$ represents the force which is proportional to the last velocity and is a kind of inertia which counteracts a change in direction. One can also write Eq. (4) as a state equation with the state vector $\boldsymbol{y} = [\boldsymbol{x} \;\; \dot{\boldsymbol{x}}]^T$, where $\boldsymbol{x}$ and $\dot{\boldsymbol{x}}$ are the translational and rotational position and velocity of the robots. Together with the initial conditions, first order of differential equation, the motion of the swarm robots over time can be computed, e.g., by the simple Euler forward integration formula, which yields the mechanical PSO model

$$\begin{aligned} \begin{bmatrix} \boldsymbol{x}^{k+1} \\ \dot{\boldsymbol{x}}^{k+1} \end{bmatrix} &= \begin{bmatrix} \boldsymbol{x}^k \\ \left(\boldsymbol{I}_{6n} - \Delta t \boldsymbol{M}^{-1}\boldsymbol{h}_{f_3}^k\right)\dot{\boldsymbol{x}}^k \end{bmatrix} \\ &+ \Delta t \begin{bmatrix} \dot{\boldsymbol{x}}^k \\ \boldsymbol{M}^{-1}\boldsymbol{h}_{f_1}^k \left(\boldsymbol{x}_{self}^{best,k} - \boldsymbol{x}^k\right) + \boldsymbol{M}^{-1}\boldsymbol{h}_{f_2}^k \left(\hat{\boldsymbol{x}}_{swarm}^{best,k} - \boldsymbol{x}^k\right) \end{bmatrix}. \end{aligned} \tag{7}$$

Comparing Eq. (7) to Eq. (3), one can see that they are quite similar and the corresponding relationships are introduced in (8), in (Tang and Eberhard, 2009) we give a very sound and useful interpretation, please see the related reference for details.

$$\Delta t \boldsymbol{M}^{-1}\boldsymbol{h}_{f_1}^k \;\longleftrightarrow\; c_1\boldsymbol{r}_1^k, \;\; \Delta t \boldsymbol{M}^{-1}\boldsymbol{h}_{f_2}^k \;\longleftrightarrow\; c_2\boldsymbol{r}_2^k, \;\; \boldsymbol{I}_{6n} - \Delta t \boldsymbol{M}^{-1}\boldsymbol{h}_{f_3}^k \;\longleftrightarrow\; \omega. \tag{8}$$

3 Design a Control Scheme for a ROBOTINO

3.1 Constraint Handling

Engineering optimization problems usually have constraints, e.g., if the particles (robots) are searching in the environment with some limitations, like interference districts or several obstacles. In this study, we also take into account the treatment of constraints. So, here the general optimization problem with an objective function and constraints is

$$\text{minimize} \quad \Psi(\boldsymbol{x}) \tag{9}$$
$$\text{subject to} \quad \begin{cases} \boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{0}, & m_e \text{ equality constraints,} \\ \boldsymbol{h}(\boldsymbol{x}) \leq \boldsymbol{0}, & m_i \text{ inequality constraints,} \end{cases}$$

where $\boldsymbol{x}$ is the position of the particle bounded additionally by $\boldsymbol{x}_{min} \leq \boldsymbol{x} \leq \boldsymbol{x}_{max}$.

For such an optimization problem, the augmented Lagrangian multiplier method can be used where each constraint violation is penalized separately by using finite penalty factors $\boldsymbol{r}_p$. Thus, the minimization problem with constraints in Eq. (9) can be transformed into an unconstrained minimization problem

$$\text{minimize} \quad L_A(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{r}_p) \tag{10}$$

$$\text{with} \quad L_A(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{r}_p) = \Psi(\boldsymbol{x}) + \sum_{i=1}^{m_e+m_i} \lambda_i P_i(\boldsymbol{x}) + \sum_{i=1}^{m_e+m_i} r_{p,i} P_i^2(\boldsymbol{x}), \quad \text{and}$$

$$P_i(\boldsymbol{x}) = \begin{cases} g_i(\boldsymbol{x}), & i = 1(1)m_e, \\ \max\left(h_{i-m_e}(\boldsymbol{x}), \frac{-\lambda_i}{2r_{p,i}}\right), & i = (m_e+1)(1)(m_e+m_i). \end{cases}$$

Please refer to (Sedlaczek and Eberhard, 2006) and (Tang and Eberhard, 2009) for details of this method.

3.2 Control Scheme

This work proposes to use a PSO based model for a swarm of mobile robots to search a target in the environment. Such a target can be specified by an objective function and several constrains, e.g., in our example

$$\text{minimize } \Psi(\boldsymbol{x}) \quad \text{with } \Psi(\boldsymbol{x}) = \frac{1}{f(\boldsymbol{x})} - \varepsilon = (x_1 - x_{m_1})^2 + (x_2 - x_{m_2})^2 \tag{11}$$

$$\text{subject to} \quad \begin{cases} h_1(\boldsymbol{x}) = 3 - x_1 \leq 0, \\ h_2(\boldsymbol{x}) = 2 - x_2 \leq 0, \\ h_3(\boldsymbol{x}) = 1 + x_1^2 - x_2^2 \leq 0. \end{cases}$$

If the center of the source $\boldsymbol{x}_m$ is infeasible, then the robots should at least get as close as possible. In this example, we choose $\boldsymbol{x}_m = (3,\ \sqrt{10})$ and the constrained minimum is $\boldsymbol{x}_{opt} = \boldsymbol{x}_m$ with $\Psi(\boldsymbol{x}_{opt}) = 0$.

We try to use our model and related algorithms in the autonomous robot ROBOTINO (see Festo Home Website, 2009) for future work, so here we extract some main components of ROBOTINO and present the control scheme for simulation and also later for practical use, see Figs. 1 and 2.

Figure 1. Components of the Festo ROBOTINO

4 Simulation and Results

Some assumptions for the simulation experiments must be made:

1. the static obstacles are represented by polygons,
2. all simulations treat only the planar case, and
3. during the avoidance of obstacles, the robots can rotate full 360 degrees.

This study classifies the simulation experiments as shown in Table 1. The symbol of a circle ‘ ○ ’ is used for a volume robot, and a ‘$*$’ for the target. The objective function and constraints are as same as described in (11).

The trajectories of five searching robots of experiment no. 1 are shown in Fig. 3, where obstacle avoidance and target searching can be seen clearly. Experiment no. 2 uses external forces to steer a robot in a specified direction, see Fig. 4. The red robot e.g. will turn left when it meets the first obstacle, see Fig. 3, but with influence from an external guiding force which is in right direction it turns to the opposite side. Compared to experiment no. 1 and no. 2, in experiment no. 3 we use more robots and try to show the ability of mutual avoidance, see Fig. 5. In experiment no. 4, the target is within an obstacle, see Fig. 6. The initial and final status are shown. As the algorithm and model are designed, the 50 robots get as close as they can to the target and siege it.

In summary, all above four experiments are obtaining the correct results from our swarm model and the designed algorithms, strategies, and methods.

5 Conclusion and Future Work

The application of PSO to our swarm model requires some extensions for the use of this scheme for the motion planning and control of large scale mobile robots. The results show that the algorithm and model used in our experiment are simple,

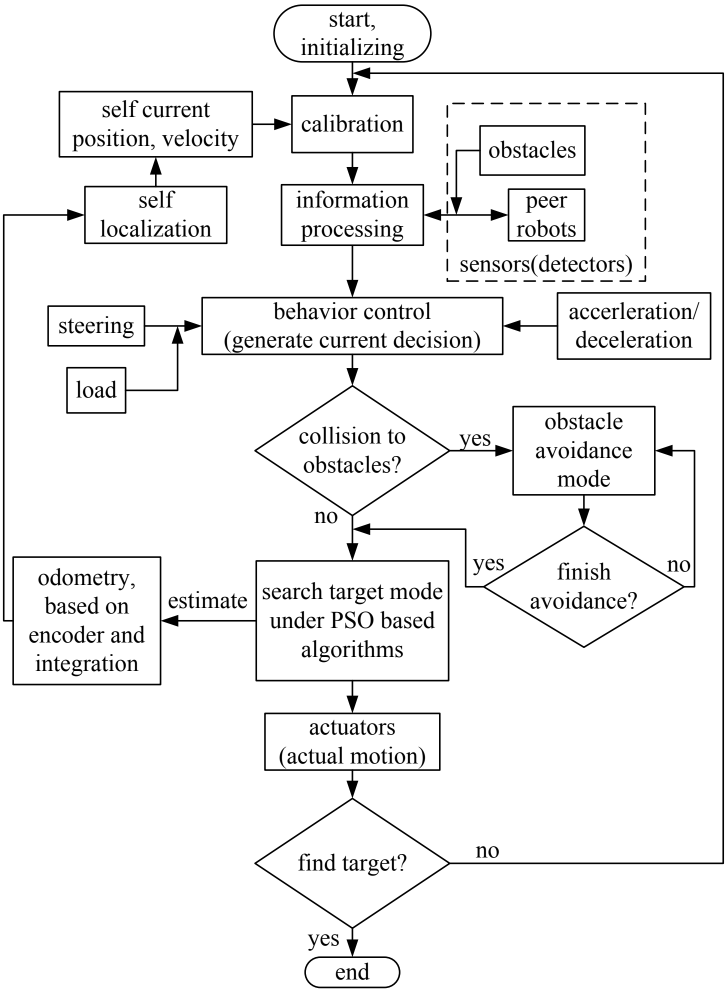

Figure 2. Control scheme

reliable, and transferable to a population of many mobile robots. It has the ability of effectively treating constraints with acceptable computational cost and it is able to avoid obstacles and other robots in the environment. The proposed model has no need of a central processor and it uses the information of each single robot's neighborhood and so it can be implemented decentralized on a large scale mobile robotic system and makes the robots move well coordinated.

Table 1. Type of experiment.

no.	robots	experiment description
1	n=5	robots coordinated movement under constraints and obstacles, target locates outside of obstacle, show trajectories
2	n=5	as experiment one, add external force to steer a robot in a specified direction
3	n=50	as experiment one, more robots, show stages
4	n=50	as experiment three, but the target is within an obstacle

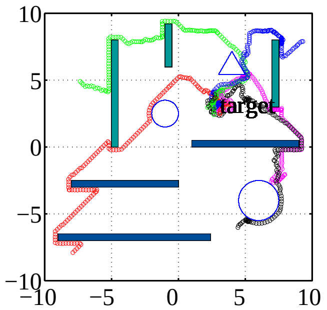

Figure 3. Result of experiment one.

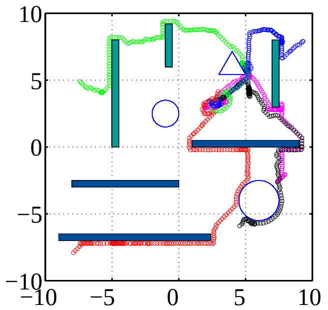

Figure 4. Result of experiment two.

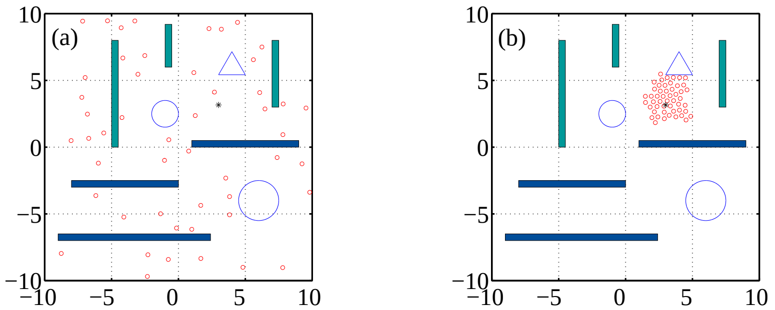

Figure 5. Experiment three, (a) t=0 s, (b) t=371.95 s.

The natural next step after the simulations are finished will be to use it for the ROBOTINO, to identify clearly the mechanical properties of a single robot and then to run everything for a population of real mobile robots.

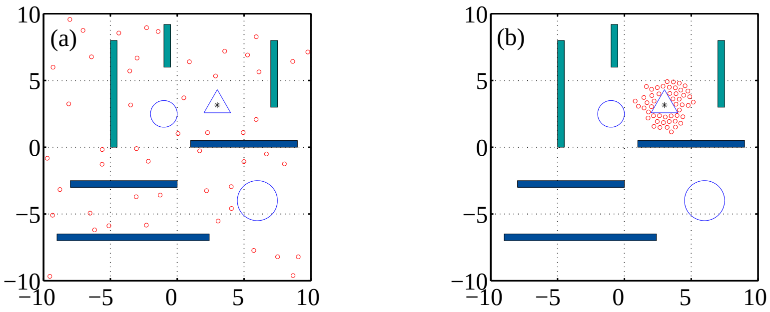

Figure 6. Experiment four, (a) t=0 s, (b) t=350.48 s.

Acknowledgements

The authors want to thank Prof. W. Schiehlen for his encouragement and detail discussion, as well as the China Scholarship Council for providing a Ph.D. fellowship to Qirong Tang to study in Germany. Parts of this research are done in the framework of SimTech, the Stuttgart Research Center for Simulation Technology. All this support is highly appreciated.

Bibliography

S. Doctor, G. K. Venayagamoorthy, and V. G. Gudise. Optimal PSO for collective robotic search applications. In *IEEE Congress on Evolutionary Computation*, pages 1390–1395, 2004.

Festo Home Website. http://www.festo-didactic.com, 2009.

J. M. Hereford and M. Siebold. Multi-robot search using a physically-embedded particle swarm optimization. *International Journal of Computational Intelligence Research*, 4(2):197–209, 2008.

J. Kennedy and R. C. Eberhart. Particle swarm optimization. In *Proceedings of the international conference on neural networks*, volume 4, pages 1942–1948, Perth, Australia, November 1995.

J. Pugh, L. Segapelli, and A. Martinoli. Applying aspects of multi-robot search to particle swarm optimization. In *Proceedings of the 5th International Workshop on Ant Colony Optimization and Swarm Intelligence*, pages 506–507, Brussels, Belgium, 2006.

K. Sedlaczek and P. Eberhard. Using augmented Lagrangian particle swarm optimization for constrained problems in engineering. *Structural and Multidisciplinary Optimization*, 32(4):277–286, 2006.

Q. R. Tang and P. Eberhard. A PSO-based algorithm designed for a swarm of mobile robots. *Structural and Multidisciplinary Optimization*, 2009. (submitted).