

Research Article

Cooperative Search by Combining Simulated and Real Robots in a Swarm under the View of Multibody System Dynamics

Qirong Tang and Peter Eberhard

Institute of Engineering and Computational Mechanics, University of Stuttgart, Pfaffenwaldring 9, 70569 Stuttgart, Germany

Correspondence should be addressed to Peter Eberhard; peter.eberhard@itm.uni-stuttgart.de

Received 26 June 2013; Accepted 13 August 2013

Academic Editor: Xiaoting Rui

Copyright © 2013 Q. Tang and P. Eberhard. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents a new approach for cooperative search of a robot swarm. After modeling the robot, the mechanical Particle Swarm Optimization method is conducted based on physical robot properties. Benefiting from the effective localization and navigation by sensor data fusion, a mixed robot swarm which contains both simulated and real robots is then successfully used for searching a target cooperatively. With the promising results from experiments based on different scenarios, the feasibility, the interaction of real and simulated robots, the fault tolerance, and also the scalability of the proposed method are investigated.

1. Introduction

Nowadays robots are used more and more since they can do various tasks instead of humans. Among the variety of existing robots, swarms of mobile robots receive growing attention by researchers. A number of large international projects have been conducted, for example, SWARM-Bots, SWARMMANOID, and SYMBARION [1]. Swarms of mobile robots have many applications, for example, for efficient collection of oil spill [2, 3], for light or odorant detection [4–6], for searching survivors after an earthquake [7], and for assisting humans to handle nuclear leakage [8]. Readers may notice from these applications that cooperative search is quite representative for the swarm robotics research since it covers most of the technical points. Many researches focus on cooperative search; however, this is still at an early stage in spite of its fast development. Many critical issues have not been solved yet, including, for example, the challenge in controlling and guiding robots systematically and the transfer to workable strategies for real robots. From a technical perspective, for cooperative search of robot swarms, key points are performing collaboration, showing fault tolerance, sharing information, distributing the work, and realizing scalability. These are also the advantages of a mobile robot swarm in contrast to a single mobile robot. Neither the traditional methods such as artificial potential fields, A^* or

D^* , nor new methods like genetic algorithms and ant colony algorithms can produce satisfactory results. In recent years, the comparatively new stochastic Particle Swarm Optimization (PSO) algorithms have attracted researchers' attention. Many research groups use PSO and its variants for swarm mobile robots search; see, for example, [9–11]. Unfortunately, these publications have shown shortcomings for cooperative search. For example, [9] used PSO for multirobot search; however, its focus is on optimizing PSO parameters and it does not consider the scalability for a large number of robots. The work in [10] gives an ad hoc application using a PSO based method for group robots search. Its solutions are valid for a specific robot which has only limited motion ability. At the moment similar researches mostly stop at a theoretical stage and are verified only by simulation. Some researchers have used PSO on real robots reluctantly, but they have limitations such as poor scalability and inadequate search ability. Thus, this study develops a systematic cooperative search method which contains sufficient search ability, scalability, and fault tolerance, considers robot physical properties, and qualifies for real-simulated robots interaction.

Section 2 builds kinematic and dynamic models of the used robot. Section 3 designs the cooperative search algorithm. Experiments performed on a mixed robot swarm for different cases are shown in Section 4, while conclusions are given in Section 5.



FIGURE 1: Festo Robotino and its main components (photo from Robotino manual [12]).

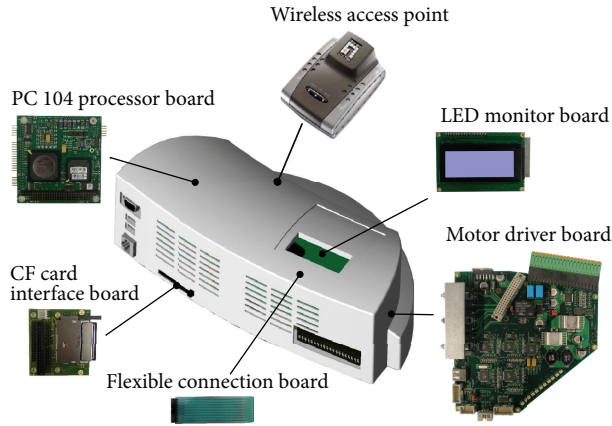


FIGURE 2: Robotino controller head and its components.

2. Kinematic and Dynamic Models

2.1. The Investigated Omnidirectional Mobile Robot. Our research object is the Festo Robotino which is an omnidirectional mobile robot. Its appearance and main components are shown in Figure 1, while Figure 2 shows the robot control unit used for realizing control and software running.

2.2. Kinematic Model. The kinematic model of a Robotino is based on the robot's structure and kinematic behavior. Since it is omnidirectional, it has the ability to move in any direction no matter what the current orientation is. This mainly results from its three special wheels; see Figure 3. The three wheel units has an angle of 120° between each other; see Figure 4. The origin of the local robot frame $[x_l \ y_l]$; see Figure 4, coincides with the center of the robot base and is set to coincide with the origin of the global frame $[x_g \ y_g]$ at the beginning but rotates the robot by angle ϕ_g . The three lateral velocities are marked as $v_{l,i}, i = 1, 2, 3$ which also can be considered as free or passive velocities. The drive forces are defined by f_1, f_2 , and f_3 . To describe the location of the three wheels relative to the local frame, we introduce the angles α_i where $\alpha_1 = 0^\circ, \alpha_2 = 120^\circ$, and $\alpha_3 = 240^\circ$. Thus, the relationship between the global velocity $[\dot{x}_g \ \dot{y}_g \ \dot{\phi}_g]$ of the

robot and the driven translational velocities v_i of three wheels is governed by

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \underbrace{\begin{bmatrix} -\sin(\phi_g + \alpha_1) & \cos(\phi_g + \alpha_1) & R \\ -\sin(\phi_g + \alpha_2) & \cos(\phi_g + \alpha_2) & R \\ -\sin(\phi_g + \alpha_3) & \cos(\phi_g + \alpha_3) & R \end{bmatrix}}_{\mathbf{S}_g} \cdot \underbrace{\begin{bmatrix} \dot{x}_g \\ \dot{y}_g \\ \dot{\phi}_g \end{bmatrix}}_{\dot{\mathbf{u}}_g}. \quad (1)$$

Here R is the distance from the robot center to the center of wheel; the matrix \mathbf{S}_g is the geometrical relationship between the global velocity $\dot{\mathbf{u}}_g$ and the translational velocity \mathbf{v} which can also be expressed by the wheel's angular velocities ω_i multiplied by the wheel radius r ; that is,

$$v_i = r\omega_i. \quad (2)$$

Inserting (2) into (1) and reformulation yields

$$\boldsymbol{\omega} = [\omega_1 \ \omega_2 \ \omega_3]^T = \frac{1}{r} \mathbf{S}_g \cdot \dot{\mathbf{u}}_g. \quad (3)$$

If \mathbf{S}_g is split into a constant part \mathbf{S}_l with the known parameters α_1, α_2 , and α_3 and a variable part $\mathbf{T}_{\text{rot}}^{-1}(\phi_g)$, the kinematic model is

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} = \frac{1}{r} \underbrace{\begin{bmatrix} 0 & 1 & R \\ -\frac{\sqrt{3}}{2} & -\frac{1}{2} & R \\ \frac{\sqrt{3}}{2} & -\frac{1}{2} & R \end{bmatrix}}_{\mathbf{S}_l} \cdot \underbrace{\begin{bmatrix} \cos(\phi_g) & \sin(\phi_g) & 0 \\ -\sin(\phi_g) & \cos(\phi_g) & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{T}_{\text{rot}}^{-1}} \cdot \underbrace{\begin{bmatrix} \dot{x}_g \\ \dot{y}_g \\ \dot{\phi}_g \end{bmatrix}}_{\dot{\mathbf{u}}_g}. \quad (4)$$

Here \mathbf{T}_{rot} is the orthogonal rotation matrix between the global and the local frames where $\mathbf{T}_{\text{rot}}^{-1} = \mathbf{T}_{\text{rot}}^T$. In addition, here $\det(\mathbf{T}_{\text{rot}}) = 1$.

2.3. Dynamic Model. Each of Robotino's three wheels is driven by its own DC motor. Wheel and motor shafts are linked by a gearbox and a gear belt with a total of 16:1 reduction ratio to enhance torque and reduce rotational velocity, see the drivetrain in Figure 5. Due to Newton's second law of motion, the planar mobile robot motion can be described by

$$\underbrace{\begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & J \end{bmatrix}}_{\mathbf{M}} \cdot \underbrace{\begin{bmatrix} \ddot{x}_g \\ \ddot{y}_g \\ \ddot{\phi}_g \end{bmatrix}}_{\ddot{\mathbf{u}}_g} = \underbrace{\begin{bmatrix} f_{g,x} \\ f_{g,y} \\ l_{g,\phi} \end{bmatrix}}_{\mathbf{f}_g}. \quad (5)$$

Here \mathbf{f}_g is the generalized global force acting on the robot, \mathbf{M} is a mass matrix where m is the mass and J is the rotational

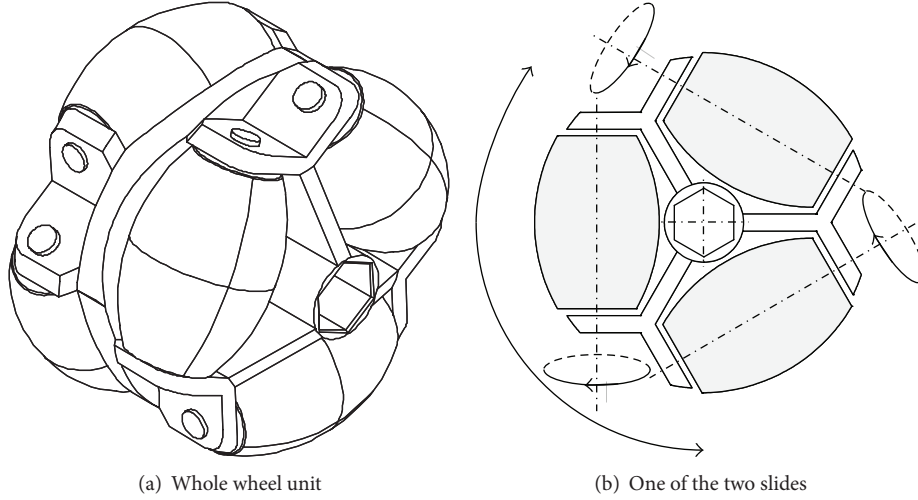


FIGURE 3: Illustration of the special wheel and its motion mechanism.

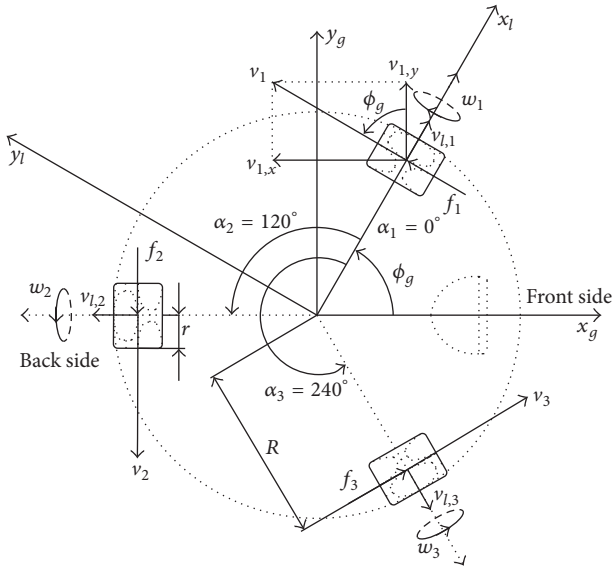


FIGURE 4: Geometry and kinematic relations of Robotino base.

moment of inertia of the robot, and $\ddot{\mathbf{u}}_g$ is the generalized acceleration. For \mathbf{f}_g and the robot local force \mathbf{f}_i one has

$$\begin{bmatrix} f_{g,x} \\ f_{g,y} \\ l_{g,\phi} \end{bmatrix} = \underbrace{\begin{bmatrix} \cos(\phi_g) & -\sin(\phi_g) & 0 \\ \sin(\phi_g) & \cos(\phi_g) & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{T}_{\text{rot}}} \cdot \begin{bmatrix} f_{i,x} \\ f_{i,y} \\ l_{i,\phi} \end{bmatrix}, \quad (6)$$

and the connection of \mathbf{f}_i to the wheel drag force \mathbf{f}_{drag} is

$$\begin{bmatrix} f_{i,x} \\ f_{i,y} \\ l_{i,\phi} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & -\frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{2} \\ 1 & -\frac{1}{2} & -\frac{1}{2} \\ R & R & R \end{bmatrix}}_{\mathbf{S}_i^T} \cdot \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} \cdot \mathbf{f}_{\text{drag}}. \quad (7)$$

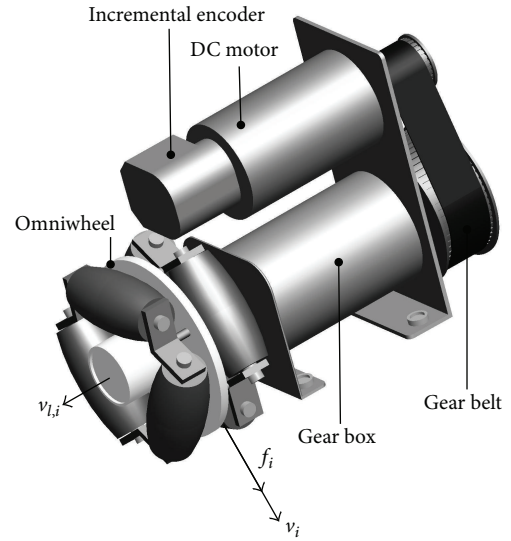


FIGURE 5: Robotino drivetrain.

Now, substituting (6) and (7) into (5) yields

$$\mathbf{f}_{\text{drag}} = (\mathbf{S}_i^T)^{-1} \cdot \mathbf{T}_{\text{rot}}^{-1} \cdot \mathbf{M} \cdot \ddot{\mathbf{u}}_g. \quad (8)$$

In the following, the Robotino's motor dynamic is investigated as well as its relation to (8). Figure 6 shows the simplified circuit diagram of the used DC motor. Based on this and Kirchhoff's voltage law, we get for the current transient changes $\dot{c}_i = dc_i/dt$ ($i = 1, 2, 3$)

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ \mathbf{U} \end{bmatrix} = R_\Omega \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \mathbf{c} \end{bmatrix} + L \begin{bmatrix} \dot{c}_1 \\ \dot{c}_2 \\ \dot{c}_3 \\ \dot{\mathbf{c}} \end{bmatrix} + \begin{bmatrix} U_{m,1} \\ U_{m,2} \\ U_{m,3} \\ \mathbf{U}_m \end{bmatrix}, \quad (9)$$

where \mathbf{U} and \mathbf{U}_m are the input voltage acting on the motors and the induced voltage from the motor winding,

respectively. The constants R_Ω and L represent resistance and inductance of the motor, respectively. According to Faraday's law of induction we have

$$\mathbf{U}_m = k_m \boldsymbol{\omega}_m, \quad (10)$$

where k_m is the back electromotive force constant which depends on the build-up of the motor, and $\boldsymbol{\omega}_m$ are the motors' angular velocities. The DC motor's torque relation is

$$\mathbf{I}_{\text{load}} = k_t \mathbf{c} - J_m \dot{\boldsymbol{\omega}}_m - k_l \boldsymbol{\omega}_m, \quad (11)$$

where k_t and k_l are the torque constant and the viscous friction coefficient, respectively. The loading torque vector \mathbf{I}_{load} is responsible for dragging the wheels, and J_m describes the motors rotational moment of inertia. Furthermore, one has

$$n \mathbf{l}_{\text{load}} = r \mathbf{f}_{\text{drag}}, \quad \boldsymbol{\omega}_m = n \boldsymbol{\omega}, \quad (12)$$

where $n = 16$ is the gear ratio. When placing (12) into (11) one obtains

$$\mathbf{f}_{\text{drag}} = \frac{k_t n}{r} \mathbf{c} - \frac{J_m n^2}{r} \dot{\boldsymbol{\omega}} - \frac{k_l n^2}{r} \boldsymbol{\omega}. \quad (13)$$

Substituting the kinematic model (4) and its time derivative

$$\dot{\boldsymbol{\omega}} = \frac{1}{r} \mathbf{S}_l \cdot (\mathbf{T}_{\text{rot}}^T)' \cdot \dot{\mathbf{u}}_g + \frac{1}{r} \mathbf{S}_l \cdot \mathbf{T}_{\text{rot}}^T \cdot \ddot{\mathbf{u}}_g, \quad (14)$$

into (13) and comparing with (8) yield the current equation

$$\begin{aligned} \frac{k_t n}{r} \mathbf{c} = & \left((\mathbf{S}_l^T)^{-1} \cdot \mathbf{T}_{\text{rot}}^T \cdot \mathbf{M} + \frac{J_m n^2}{r^2} \mathbf{S}_l \cdot \mathbf{T}_{\text{rot}}^T \right) \cdot \dot{\mathbf{u}}_g \\ & + \left(\frac{J_m n^2}{r^2} \mathbf{S}_l \cdot (\mathbf{T}_{\text{rot}}^T)' + \frac{k_l n^2}{r^2} \mathbf{S}_l \cdot \mathbf{T}_{\text{rot}}^T \right) \cdot \mathbf{u}_g. \end{aligned} \quad (15)$$

Undetermined parameters in (15) can be read or calculated from the motors characteristics in Figure 7. Thus, with (15) the required current and further the actual applied voltage are calculated for resulting robot motions.

3. Design of the Cooperative Search Algorithm Based on Particle Swarm Optimization

3.1. The Basic Particle Swarm Optimization Model. Particle Swarm Optimization (PSO) is usually considered to be inspired from swarms of birds and schools of fish. For example, in Figure 8 a swarm of birds is shown in which each bird is taken as an adaptive agent and can get information from the environment and other agents.

The computational algorithm of PSO was first introduced by Kennedy and Eberhart in [14]. In PSO, a swarm consists of N_p members called particles, and each possesses information about position and velocity in the work space. The particles are generated with an initial random position and velocity within a defined range. The recursions of basic PSO use the vectors $\Delta \mathbf{x}$ and \mathbf{x} to denote the particles "velocity" and

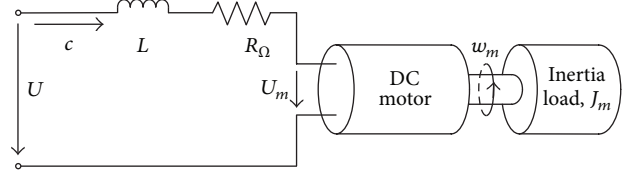


FIGURE 6: Circuit diagram of the DC motor used in the Robotino.

actual position, respectively. The so-called "velocity" of the i th particle at the $(s + 1)$ th iteration can be updated by

$$\Delta \mathbf{x}_i^{s+1} = \omega \Delta \mathbf{x}_i^s + c_1 r_{i,1}^s (\mathbf{x}_{i,\text{self}}^{\text{best},s} - \mathbf{x}_i^s) + c_2 r_{i,2}^s (\mathbf{x}_{\text{swarm}}^{\text{best},s} - \mathbf{x}_i^s), \quad (16)$$

and the position update is done by

$$\mathbf{x}_i^{s+1} = \mathbf{x}_i^s + \Delta \mathbf{x}_i^{s+1}. \quad (17)$$

The term $\omega \Delta \mathbf{x}_i^s$ in (16) is the inertia part which contains the coefficient ω . The factors c_1 as well as c_2 in (16) are used for cognitive and social scaling which are generated from the attractions to the previous personal best position $\mathbf{x}_{i,\text{self}}^{\text{best},s}$ and the entire swarm best position $\mathbf{x}_{\text{swarm}}^{\text{best},s}$, respectively. The factors $r_{i,1}^s$ and $r_{i,2}^s$ are uniformly distributed random variables both range in $[0, 1]$. In (17), the "velocity" actually corresponds to a displacement $\Delta \mathbf{x}_i^{s+1} = h \dot{\mathbf{x}}_i^s$ where h is the time step and $\dot{\mathbf{x}}_i^s$ is the mechanical velocity. However, in PSO literature h is usually set to be one second. Thus, for all N_p particles, the basic PSO algorithm may be summarized as

$$\begin{aligned} \begin{bmatrix} \mathbf{x}^{s+1} \\ \dot{\mathbf{x}}^{s+1} \end{bmatrix} = & \begin{bmatrix} \mathbf{x}^s \\ \omega \dot{\mathbf{x}}^s \end{bmatrix} \\ & + \begin{bmatrix} c_1 \mathbf{r}_1^s \cdot (\mathbf{x}_{\text{self}}^{\text{best},s} - \mathbf{x}^s) + c_2 \mathbf{r}_2^s \cdot (\mathbf{x}_{\text{swarm}}^{\text{best},s} - \mathbf{x}^s) \end{bmatrix}, \end{aligned} \quad (18)$$

where

$$\mathbf{r}_i^s = \text{diag}(r_{1,i}^s \mathbf{I}_q, r_{2,i}^s \mathbf{I}_q, \dots, r_{N_p,i}^s \mathbf{I}_q), \quad i \in \{1, 2\} \quad (19)$$

are generated independently for every iteration. Here, q is the generalized dimension. Considering the case with 3 generalized dimensions, $\mathbf{I}_q = \mathbf{I}_3$ is a 3×3 unit matrix. Similarly the matrices

$$\mathbf{x}_i^s = \begin{bmatrix} x_{1,i}^s \\ x_{2,i}^s \\ x_{3,i}^s \end{bmatrix}, \quad \mathbf{x}_{i,\text{self}}^{\text{best},s} = \begin{bmatrix} x_{1,i,\text{self}}^{\text{best},s} \\ x_{2,i,\text{self}}^{\text{best},s} \\ x_{3,i,\text{self}}^{\text{best},s} \end{bmatrix},$$

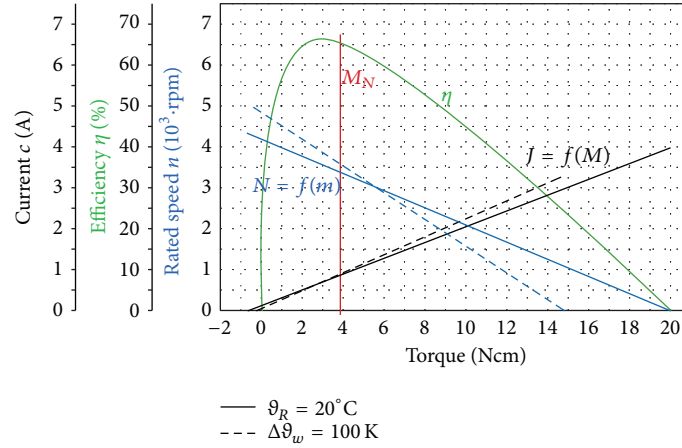


FIGURE 7: Characteristic diagram of the DC motor-GR 42 × 25 DC motor used on Robotino (data from information of Dunkermotoren [13]).



FIGURE 8: Swarm of birds (photo taken at Seine riverside in Paris, France, 2011).

3.2. Extension to Mechanical Particle Swarm Optimization. In recent years, the PSO has shown its distinctive advantages for guiding robot swarm motions, for example, in cooperative search. The particles in PSO share some information during the search according to their update formulae and try to find the minimum of an objective function. This is quite similar to a robot swarm searching for a target in an environment according to some search algorithms. If one considers that each particle represents one robot, then the particle motions are determined not only by the PSO update but also by robots mechanical properties and their drives. Unfortunately, the PSO itself is only an optimization tool. Many researches like [4, 9, 11, 16] work without much attention to the robot properties, not to mention including the properties into the guiding algorithms. Most of them are still using PSO as an optimization tool during the search processes. It can be predicted that if they perform motions with real robots, extensive control is required to realize the swarm motions and to solve the difficulties caused by the ill consideration.

Actually, the physical properties of each member in the robotic swarm can seriously affect the performance of the whole swarm system since each individual is a mechatronical system. Thus, the physical background and physical laws must be considered to generate reasonable search trajectories. In this work, we trace back to the PSO algorithm and consider the mechanical properties of robots such as mass, inertia, and forces. This is a big difference between our research and others', which results in our algorithm named mechanical Particle Swarm Optimization. In addition, we assume that the required forces on a robot in mechanical PSO are caused by other robots; that is, the whole robot swarm is a virtually linked multibody system. Physically, the forces are created by corresponding robot actuators.

Considering a real mobile robot i , it moves in a time step Δt by

$$\begin{bmatrix} \mathbf{x}_i^{s+1} \\ \dot{\mathbf{x}}_i^{s+1} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_i^s \\ \dot{\mathbf{x}}_i^s \end{bmatrix} + \Delta t \begin{bmatrix} \dot{\mathbf{x}}_i^s \\ \ddot{\mathbf{x}}_i^s \end{bmatrix}, \quad (21)$$

where the robot position \mathbf{x}_i^s is given in absolute coordinates, and the simple Euler forward integration method is used.

$$\mathbf{x}_{\text{swarm}}^{\text{best},s} = \begin{bmatrix} \mathbf{x}_{1,\text{swarm}}^{\text{best},s} \\ \mathbf{x}_{2,\text{swarm}}^{\text{best},s} \\ \mathbf{x}_{3,\text{swarm}}^{\text{best},s} \end{bmatrix}, \quad i = 1, 2, \dots, N_p,$$

$$\mathbf{x}^s = \begin{bmatrix} \mathbf{x}_1^s \\ \mathbf{x}_2^s \\ \vdots \\ \mathbf{x}_i^s \\ \vdots \\ \mathbf{x}_{N_p}^s \end{bmatrix}, \quad \mathbf{x}_{\text{self}}^{\text{best},s} = \begin{bmatrix} \mathbf{x}_{1,\text{self}}^{\text{best},s} \\ \mathbf{x}_{2,\text{self}}^{\text{best},s} \\ \vdots \\ \mathbf{x}_{i,\text{self}}^{\text{best},s} \\ \vdots \\ \mathbf{x}_{N_p,\text{self}}^{\text{best},s} \end{bmatrix},$$

$$\hat{\mathbf{x}}_{\text{swarm}}^{\text{best},s} = \begin{bmatrix} \mathbf{x}_{\text{swarm}}^{\text{best},s} \\ \mathbf{x}_{\text{swarm}}^{\text{best},s} \\ \vdots \\ \mathbf{x}_{\text{swarm}}^{\text{best},s} \end{bmatrix}, \quad \mathbf{x}^s, \mathbf{x}_{\text{self}}^{\text{best},s}, \hat{\mathbf{x}}_{\text{swarm}}^{\text{best},s} \in \mathbb{R}^{3N_p \times 1} \quad (20)$$

are defined. For more details one can refer to [15].

Here $\ddot{\mathbf{x}}_i^s$ is the generalized acceleration of robot i at time step s , and $\dot{\mathbf{x}}_i^s$ is its velocity.

In multibody dynamics, one knows that all the external influences acting on a rigid body i can be synthesized by resultant forces and torques when our robots work on a plane ground. Due to the Newton-Euler equations, the motion of robot i can be described by

$$\begin{bmatrix} m_i & 0 & 0 \\ 0 & m_i & 0 \\ 0 & 0 & J_i \end{bmatrix} \cdot \begin{bmatrix} a_{i,x} \\ a_{i,y} \\ \alpha_i \end{bmatrix} = \begin{bmatrix} f_{i,x} \\ f_{i,y} \\ l_i \end{bmatrix}, \quad (22)$$

where J_i is the rotational moment of inertia, $a_{i,x}$ and $a_{i,y}$ are the resultant accelerations in x and y directions, respectively, while $f_{i,x}$ and $f_{i,y}$ are the corresponding forces. The α_i represents the planar rotational acceleration, and l_i is the corresponding torque. Thus, the generalized planar acceleration for robot i can be written as

$$\ddot{\mathbf{x}}_i = [a_{i,x} \ a_{i,y} \ \alpha_i]^T. \quad (23)$$

If one defines the generalized acceleration

$$\ddot{\mathbf{x}} = [\ddot{\mathbf{x}}_1 \ \ddot{\mathbf{x}}_2 \ \cdots \ \ddot{\mathbf{x}}_i \ \cdots \ \ddot{\mathbf{x}}_{N_p}]^T \in \mathbb{R}^{3N_p \times 1} \quad (24)$$

for all N_p robots and \mathbf{k} as the term resulting from Euler equations, as well as a definition that \mathbf{q} contains the information of forces and moments, then the motion of the entire swarm is described by

$$\mathbf{M} \cdot \ddot{\mathbf{x}} + \mathbf{k} = \mathbf{q} \quad \text{or} \quad \ddot{\mathbf{x}} = \mathbf{M}^{-1} \cdot (\mathbf{q} - \mathbf{k}) = \mathbf{M}^{-1} \cdot \mathbf{F}, \quad (25)$$

where \mathbf{F} contains the generalized forces. Here, the multibody system is a free system since no links or joints are involved. Thus,

$$\mathbf{M} = \text{diag}(m_1, m_1, J_1, m_2, m_2, J_2, \dots, m_{N_p}, m_{N_p}, J_{N_p}) \quad (26)$$

possesses both the information of masses and rotational inertias of all the robots in the swarm. With (21) and (25), the motion of all N_p robots over time can be computed by

$$\begin{bmatrix} \mathbf{x}^{s+1} \\ \dot{\mathbf{x}}^{s+1} \end{bmatrix} = \begin{bmatrix} \mathbf{x}^s \\ \dot{\mathbf{x}}^s \end{bmatrix} + \Delta t \begin{bmatrix} \dot{\mathbf{x}}^s \\ \mathbf{M}^{-1} \cdot \mathbf{F}^s \end{bmatrix}. \quad (27)$$

The used simple Euler forward algorithm is neither very stable nor very efficient. However, these aspects are not critical here, because for signals provided from sensors of real robots, the introduced inaccuracies are even worse.

Next, the essential connection between the mechanical motion of robots and the mechanical PSO will be made. At the moment, the robots are only affected by forces such that no torques appear from other robots; that is, $l_i = 0$. Of course the robot itself can provide its own torques to perform rotations. In other words, the trajectories generated from mechanical PSO only include the two translational motions. There are mainly two reasons to design it like this. One important reason is that we want to benefit from the omnidirectional feature of the robot so as to reduce the

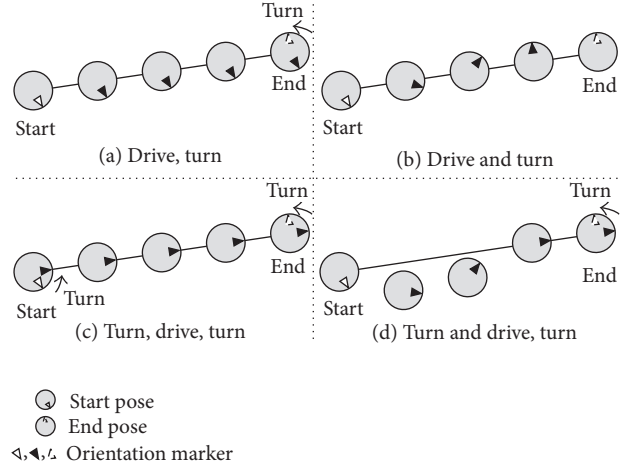


FIGURE 9: Robot omnidirectional feature.

consumption of energy. In Figure 9 one can see four basic motion modes for the robot which starts from its initial pose to the desired final pose not strictly along a preferred trajectory.

The motions in Figures 9(a) and 9(b) are preferred since our researched robot is omnidirectional, and these two modes can save energy. Nonetheless, it is still difficult to run mode (b) directly due to the challenges met in robot position control since the nonlinearity of this robot motion becomes significant when the robot orientation ϕ_g is changing simultaneously. The reason behind this is that the orientation ϕ_g is contained in many entries of the system matrix when describing this robot model in the state space for designing proper controllers. The changing orientation will play an important role in contributing the nonlinearity. Thus, we choose mode (a) as a trade-off using the omnidirectional feature of the robot. The whole trajectory is formed by many small segments. Although the robot does not have the torque affection from other robots, some additional rotation adjustments from the robot itself still can be performed if necessary. In analogy to (18), the force \mathbf{F}^s is assumed to consist of three parts,

$$\begin{aligned} \mathbf{f}_1^s &= -\mathbf{h}_{f_1}^s \cdot (\mathbf{x}^s - \mathbf{x}_{\text{self}}^{\text{best},s}), \\ \mathbf{f}_2^s &= -\mathbf{h}_{f_2}^s \cdot (\mathbf{x}^s - \hat{\mathbf{x}}_{\text{swarm}}^{\text{best},s}), \quad \mathbf{f}_3^s = -\mathbf{h}_{f_3}^s \cdot \dot{\mathbf{x}}^s, \end{aligned} \quad (28)$$

where $\mathbf{h}_{f_1}^s$, $\mathbf{h}_{f_2}^s$, and $\mathbf{h}_{f_3}^s$ are force factor matrices. Here, the swarm mode $\hat{\mathbf{x}}_{\text{swarm}}^{\text{best},s}$ can be adjusted to a neighborhood mode $\mathbf{x}_{\text{nhood}}^{\text{best},s}$ when the swarm size is big which brings difficulties for the robots communication. Force \mathbf{f}_1^s has the physical meaning of an attraction force by the last self best position, \mathbf{f}_2^s represents the attraction force by neighborhood or swarm best position, and \mathbf{f}_3^s is a kind of inertia force which is proportional to its last velocity and counteracts a change in

direction. Thus, the motion (27) of the entire swarm robotic system is governed by

$$\begin{aligned} \begin{bmatrix} \mathbf{x}^{s+1} \\ \dot{\mathbf{x}}^{s+1} \end{bmatrix} &= \begin{bmatrix} \mathbf{x}^s \\ (\mathbf{I}_{3N_p} - \Delta t \mathbf{M}^{-1} \cdot \mathbf{h}_{f_3}^s) \cdot \dot{\mathbf{x}}^s \end{bmatrix} \\ &+ \Delta t \begin{bmatrix} \dot{\mathbf{x}}^s \\ \mathbf{M}^{-1} \cdot \mathbf{h}_{f_1}^s \cdot (\mathbf{x}_{\text{self}}^{\text{best},s} - \mathbf{x}^s) \\ + \mathbf{M}^{-1} \cdot \mathbf{h}_{f_2}^s \cdot (\hat{\mathbf{x}}_{\text{swarm}}^{\text{best},s} - \mathbf{x}^s) \end{bmatrix}. \end{aligned} \quad (29)$$

Comparing the developed mechanical PSO model (29) for generating trajectories for swarm mobile robots cooperative search to the basic PSO model (18), one can see corresponding relationships as follows:

$$\begin{aligned} \Delta t \mathbf{M}^{-1} \cdot \mathbf{h}_{f_1}^s &\longleftrightarrow c_1 \mathbf{r}_1^s, & \Delta t \mathbf{M}^{-1} \cdot \mathbf{h}_{f_2}^s &\longleftrightarrow c_2 \mathbf{r}_2^s, \\ \mathbf{I}_{3N_p} - \Delta t \mathbf{M}^{-1} \cdot \mathbf{h}_{f_3}^s &\longleftrightarrow \omega. \end{aligned} \quad (30)$$

The mass matrix \mathbf{M} can be different and can even be changed during motion for each robot which enhances the flexibility. Random effects are included in $\mathbf{h}_{f_1}^s$, $\mathbf{h}_{f_2}^s$, and $\mathbf{h}_{f_3}^s$, and all forces must be created equivalently by local actuators in the robots.

4. Experiments with a Mixed Robot Swarm

4.1. General Experiment Specifications. The experimental environment is a 3 m × 3 m square, and the origin of the coordinate system is set at the center. Thus, both for x - and y -directions the experimental range is $[-1.5 \text{ m } 1.5 \text{ m}]$; see Figure 10. Obstacles and robots in the real environment have a one-to-one counterpart in the simulation. We use a red circle for representing a simulated robot and a gray circle for the one which uses position and velocity information from a real physical robot. The searched target is set at $(-0.5 \text{ m}, 0.45 \text{ m})$ and represented by a red pentagon. However, the robots do not know it. Five real obstacles are also shown in Figure 10. The searching robots are guided by mechanical PSO step by step; that is, the robots are tracking the mechanical PSO generated trajectories online based on the derived robot kinematic and dynamic models under the help of robot position control [17]. An obstacle avoidance module which is able to avoid both robot-obstacle and robot-robot collisions is activated if there is the danger of collisions. The obstacle avoidance module has a higher priority than mechanical PSO; that is, the latter should be collision free. Worth to be mentioned is that the simulation and real experiment are synchronized which means we can realize simulated robots and real robots in a mixed search simultaneously.

Since the robots are searching for a target cooperatively in the environment and the target is described in a general way, the performance of the fitness function in mechanical PSO is represented by the distances between current robots and target. Nevertheless, the target is unknown to the robots. The only change required for instantiation is to change the performance type in the fitness function. For example, in the case of light source search, the distances should be replaced by the strength of the detected light from each robot.

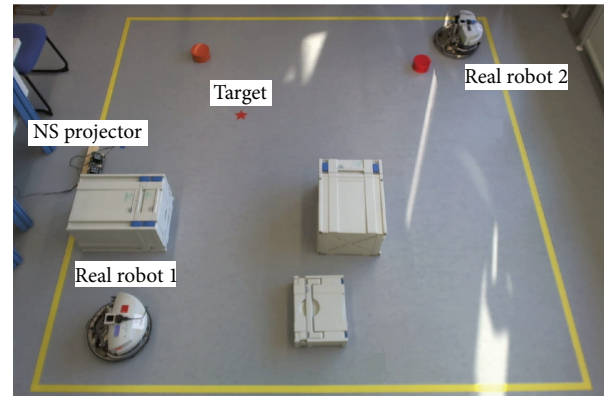
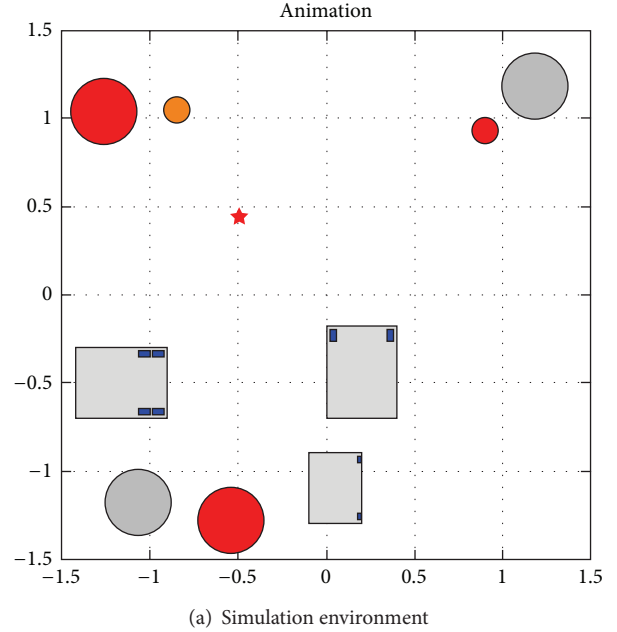


FIGURE 10: Environment for robot swarm search.

Initial robot poses can be changed and can be arbitrary. However, the pose is required by mechanical PSO for its update. Therefore, the evaluation center should be informed about the initial pose although the robots themselves do not need to know. If the robot has the ability to self-localize its initial position with sufficient accuracy, the user can randomly place the robot to the searched environment without the necessity of manually measuring its pose. In this investigation we directly measure the global initial pose of the “randomly placed robot” since the used robot lacks a sufficiently accurate self-localization of its initial pose. However, our mechanical PSO is designed for robots with arbitrary initial poses.

4.2. Verify Real-Simulated Interactions and Fault Tolerance. This experiment focuses on the real-simulated robots interaction and the fault tolerance ability of the developed mechanical PSO for robots cooperative search. Thus, during the search process, one simulated robot is artificially set to lose

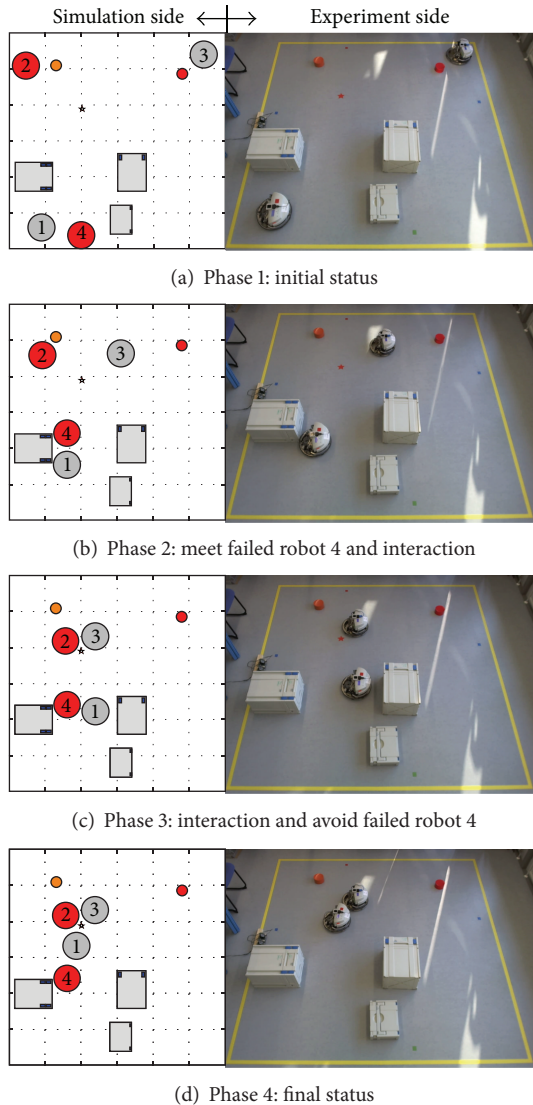


FIGURE 11: Four critical phases of an interaction and fault tolerance experiment.

its wireless signal and stop there. Totally four robots with two real robots 1 and 3 and two simulated ones 2 and 4 are used in this experiment.

Four representative states, that is, the initial status, the status when robot 1 meets and interacts with the simulated failed robot 4, the status when robot 1 avoids the failed robot and passes the challenging narrow way, and the final status of one of the experiments are demonstrated in Figure 11. The robot group uses 189 s for the pure search process, that is, without counting sensor execution and delay time from real robots. From Figure 11 one can see that robot 4 fails after phase 2 during the mission. It stops at one of the obstacle's corners. This leads to the interactions between the real robot 1 and the simulated robot 4. Robot 1 needs to avoid robot 4 while keeping collision free to the obstacles at its left and right sides. This is a big challenge for robot 1 since it needs to pass a narrow channel due to the blocking from the

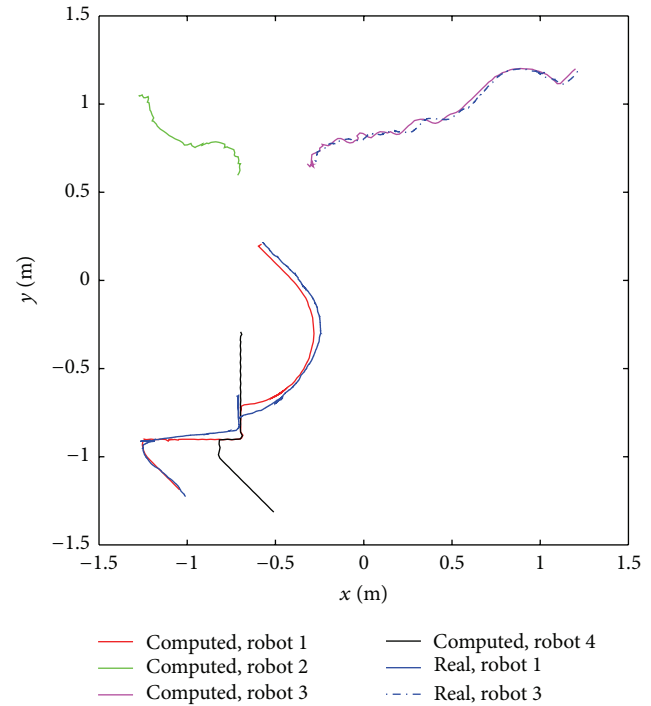


FIGURE 12: Robot trajectories of an interaction and fault tolerance experiment.

failed robot. In this case the mechanical PSO should have an affordable search ability, and the real robot should have sufficient localization accuracy.

Robot 1 and robot 3 have real-real interactions when they are close to the target. We ran this kind of experiment many times, and positive results were obtained. Some of the trajectories are shown in Figure 12 with the computed and real trajectories being compared. From the convincing trajectory figure one can see that our developed mechanical PSO is valid for such a challenging case. Robot 1 successfully passes the narrow way and approaches the target while it does not collide to the failed robot and the obstacles nearby.

The robots interaction, fault tolerance, and mutual avoidance are also shown easily by the velocity curves in Figures 13 and 14 for robot 1 and robot 4, respectively. Both figures have a good agreement to the trajectories in Figure 12. From Figure 13 one can see that robot 1 has several critical time domains (r1-r4 marked with green circles) where this real robot interacts with the failed robot and avoids obstacles. In the area of r1, robot 1 and robot 4 have the first interaction since the velocities in x - and y -directions of robot 1 go through the zero point. In the r2 area, robot 1 has two very short time interactions with robot 4 since before that moment robot 1 moves at positive x -direction along one obstacle while robot 4 meets the same obstacle (left box) and tries to move at positive y -direction with almost zero velocity in x -direction after “negotiating” with the obstacle. The two robots then meet at the obstacle corner and the interaction happens. At the area of r3, robot 4 has already stopped for a while, so robot 1 has to run a negative y velocity since it was following robot 4

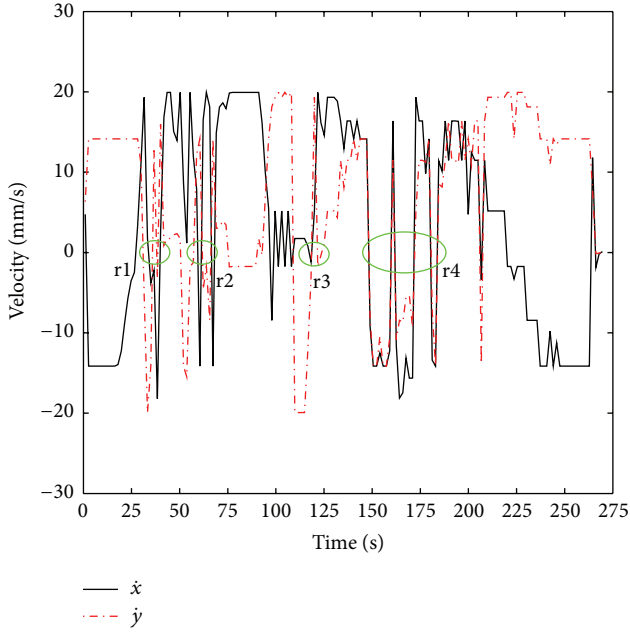


FIGURE 13: Robot 1 global velocity change.

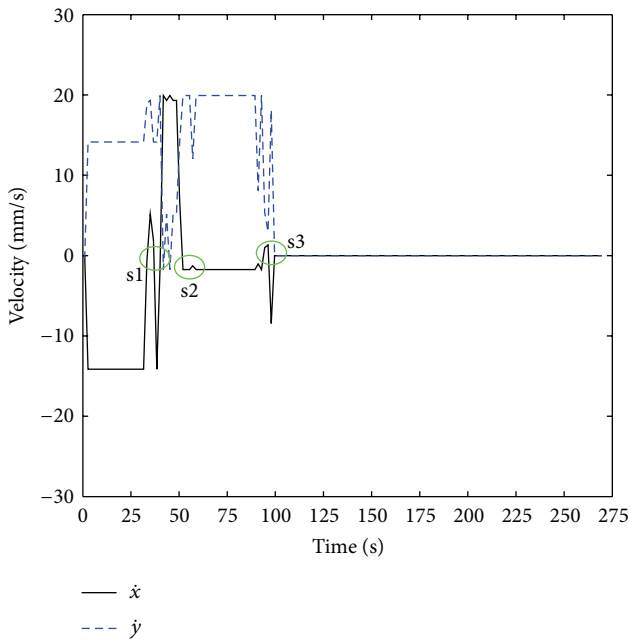
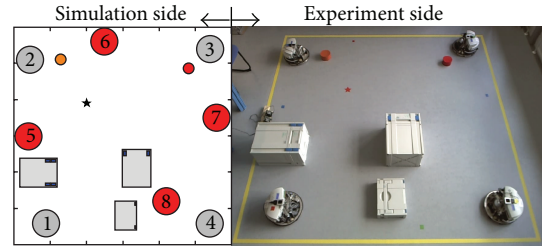
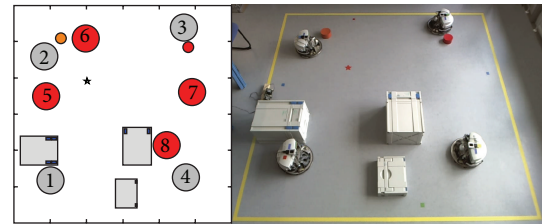


FIGURE 14: Robot 4 global velocity change.

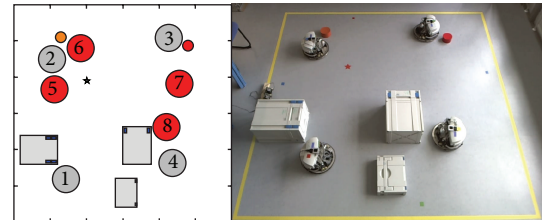
along the vertical boundary of the nearby obstacle (left box). Thus, the time of r3 is also a little bit delayed compared to the time of s3. In the area of r4, robot 1 is trying to pass the challenging narrow channel; thus it has several fluctuations. Accordingly, robot 4 has the s1-s3 feature points. Robot 4 is artificially set to fail after about 100 s running. From Figures 13 and 14 we also can see that the velocity of the simulated robot is much smoother than that of the real robot. This is mainly due to the slippage of the real robot.



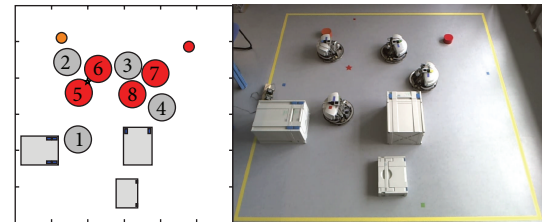
(a) Phase 1: initial status



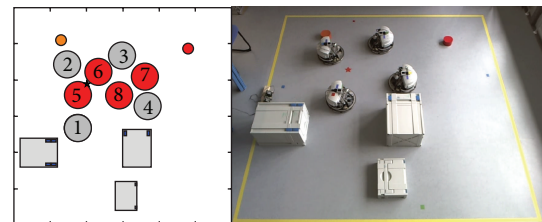
(b) Phase 2: handling obstacles



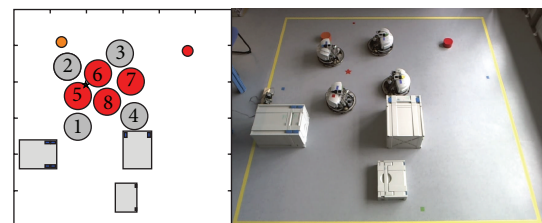
(c) Phase 3: interaction at left up corner



(d) Phase 4: all robots avoided obstacles



(e) Phase 5: swarm cooperation



(f) Phase 6: final status

FIGURE 15: Critical phases of one of the runs with mixed simulated and real robots in a swarm.

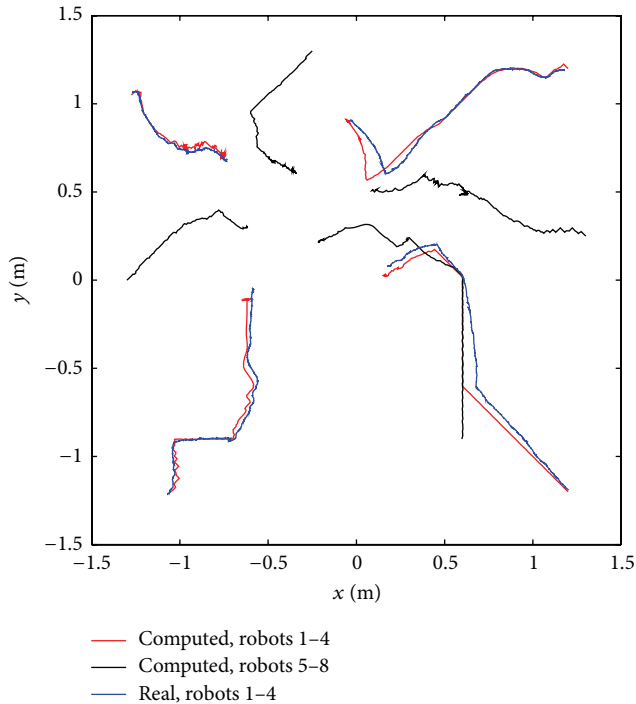


FIGURE 16: Trajectories of one of the runs with mixed simulated and real robots.

Another very interesting feature worth to be emphasized is the powerful search ability of mechanical PSO. After the failure of robot 4, only three robots remain as “particle swarm” and still possess the swarm advantage. This confirms that our method for the robots cooperative search is running in a systematic way and is not completely dependent on one or several of the robots.

4.3. Cooperative Search by a Robot Swarm Mixed with Simulated and Real Robots. A very natural further step is to increase the number of robots for cooperative search, that is, to set up a bigger robot swarm which is also mixed of simulated and real robots. This is a comprehensive verification; besides the feasibility, it also aims to verify the scalability and flexibility of the investigated methods. In this experiment, four simulated robots 5–8 and four real robots 1–4 are used which are marked with numbers in Figure 15. The number of used robots is quite flexible; however, due to spatial restrictions in this laboratory scenario, only eight robots are included here.

Some important phases of one of the runs of this mixed swarm cooperative search experiment are presented in Figure 15. The initial phase of this run is shown in Figure 15(a) where eight robots are distributed in the search environment. In the second phase, most of the robots meet or yet get closer to obstacles, and these robots are trying to deal with nearby obstacles; see Figure 15(b).

The first interaction between simulated and real robots happens at the left-upper corner in Figure 15(c). In phase 4, Figure 15(d), one can see that all the robots successfully

avoid obstacles and approach the target. Swarm cooperation is basically demonstrated in Figure 15(e) where the robots are performing coordination for finding a reasonable position. Eventually, after consuming 165 s for the search algorithm, its final status is recorded in phase 6; see Figure 15(f).

For this robot swarm cooperative search experiment, trajectories of robots are also recorded in Figure 16 which further confirms the applicability of the developed methods.

Comparing the two experiments, the second one increases the number of robots for both simulated and real robots. However, the pure search time is reduced in the second experiment since more robots have a better search ability. But due to the real robot execution and sensor delay time, the total time used for the second experiment is slightly increased. Nevertheless, as long as the robot number increases further, this trend will disappear since our solution is partially distributed.

By performing search experiments with a mixed set of simulated and real robots, the purpose of this investigation is verified. It is a good choice to enhance the search ability of the robot swarm by this method. Furthermore, it provides a nice way for the investigation of simulation and practical experiments interactively; that is, it bridges tightly the simulation side and the experiment side.

5. Conclusions

This work investigates cooperative search of a robot swarm. Kinematic and dynamic modeling of the used robot is performed to provide the basis for the control plant. Then the mechanical PSO algorithm which includes robot physical properties is described for generating search trajectories of the robots. This gives a powerful search ability. Those considered forces between the robot members are used in a virtually linked multibody system. With convincing results from experiments, the goal of hardware swarm mobile robots searching for a target collaboratively is achieved. Performing cooperative search by mixed simulated and real robots in a swarm is another very challenging investigation of this work and is one of its novelties. It shows the scalability and flexibility of a robot swarm. The whole scheme was instantiated in a planar case; however, our method can be extended in the future to a 3D case and to include torques as well. In addition, the guiding mechanical PSO method itself does not rely on specific models of the used robots. Thus, our guiding scheme is qualified also for heterogeneous robot models. According to the obtained encouraging results from experiments, one can claim that the investigated method shows its validity, feasibility, reliability (fault tolerance), qualification for real-simulated robots interaction, and scalability. Based on these, many further experiments are currently done.

Acknowledgments

Part of this work is supported by the Cluster of Excellence Simulation Technology (SimTech) in Stuttgart, Germany, which is greatly acknowledged. The publication is supported

by the German Research Foundation (DFG) within the funding programme Open Access Publishing. All these supports are highly appreciated.

References

- [1] R. Bogue, "Swarm intelligence and robotics," *Industrial Robot*, vol. 35, no. 6, pp. 488–495, 2008.
- [2] N. M. P. Kakalis and Y. Ventikos, "Robotic swarm concept for efficient oil spill confrontation," *Journal of Hazardous Materials*, vol. 154, no. 1–3, pp. 880–887, 2008.
- [3] D. Fritsch, *Steuerung selbstorganisierender Multi-Roboter-Systeme für dynamische Sammelaufgaben am Beispiel der Bekämpfung maritimer Ölverschmutzungen [Ph.D. thesis]*, University of Stuttgart, Stuttgart, Germany, 2009.
- [4] J. M. Hereford, M. Siebold, and S. Nichols, "Using the particle swarm optimization algorithm for robotic search applications," in *Proceedings of the IEEE Swarm Intelligence Symposium (SIS '07)*, pp. 53–59, Honolulu, Hawaii, USA, April 2007.
- [5] A. T. Hayes, A. Martinoli, and R. M. Goodman, "Swarm robotic odor localization: off-line optimization and validation with real robots," *Robotica*, vol. 21, no. 4, pp. 427–441, 2003.
- [6] W. Jatmiko, K. Sekiyama, and T. Fukuda, "A PSO-based mobile robot for odor source localization in dynamic advection-diffusion with obstacles environment: theory, simulation and measurement," *IEEE Computational Intelligence Magazine*, vol. 2, no. 2, pp. 37–51, 2007.
- [7] E. Rohmer, T. Yoshida, K. Ohno, K. Nagatani, S. Tadokoro, and E. Konayagi, "Quince: a collaborative mobile robotic platform for rescue robots research and development," in *Proceedings of the 5th International Conference on Advanced Mechatronics*, pp. 225–230, Osaka, Japan, October 2010.
- [8] R. Quinn, "US Sending Robots to Fukushima," 2011, <http://www.newser.com>.
- [9] S. Doctor, G. K. Venayagamoorthy, and V. G. Gudise, "Optimal PSO for collective robotic search applications," in *Proceedings of the Congress on Evolutionary Computation (CEC '04)*, pp. 1390–1395, June 2004.
- [10] J. M. Hereford and M. Siebold, "Multi-robot search using a physically embedded Particle Swarm Optimization," *International Journal of Computational Intelligence Research*, vol. 4, no. 2, pp. 197–209, 2008.
- [11] J. Pugh, L. Segapelli, and A. Martinoli, "Applying aspects of multirobot search to Particle Swarm Optimization," in *Proceedings of the 5th International Workshop on Ant Colony Optimization and Swarm Intelligence*, pp. 506–507, Brussels, Belgium, September 2006.
- [12] M. Bliesener, C. Weber, K. Kling, U. Karras, and D. Zitzmann, *Festo Robotino Manual*, Festo Didactic GmbH & Co. KG, Denkendorf, Germany, 2007.
- [13] Dunkermotoren, *Permanent Magnet DC-Motors*, 2009.
- [14] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, December 1995.
- [15] Q. Tang and P. Eberhard, "A PSO-based algorithm designed for a swarm of mobile robots," *Structural and Multidisciplinary Optimization*, vol. 44, no. 4, pp. 483–498, 2011.
- [16] S. Xue and J. Zeng, "Controlling swarm robots for target search in parallel and asynchronously," *International Journal of Modelling, Identification and Control*, vol. 8, no. 4, pp. 353–360, 2009.
- [17] P. Eberhard and Q. Tang, "Sensor data fusion for the localization and position control of one kind of omnidirectional mobile robots," in *Multibody System Dynamics, Robotics and Control*, J. Gerstmayr and H. Gatringer, Eds., pp. 45–74, Springer, Wien, Austria, 2013.