Mechanical PSO Aided by Extremum Seeking for Swarm Robots Cooperative Search

Qirong Tang and Peter Eberhard

Institute of Engineering and Computational Mechanics,
University of Stuttgart, Pfaffenwaldring 9, 70569 Stuttgart, Germany
[qirong.tang; peter.eberhard]@itm.uni-stuttgart.de

Abstract. This paper addresses the issue of swarm robots cooperative search. A swarm intelligence based algorithm, mechanical Particle Swarm Optimization (PSO), is first conducted which takes into account the robot mechanical properties and guiding the robots searching for a target. In order to avoid the robot localization and to avoid noise due to feedback and measurements, a new scheme which uses Extremum Seeking (ES) to aid mechanical PSO is designed. The ES based method is capable of driving robots to the purposed states generated by mechanical PSO without the necessity of robot localization. By this way, the whole robot swarm approaches the searched target cooperatively. This pilot study is verified by numerical experiments in which different robot sensors are mimicked.

Keywords: Swarm Robotics, Mechanical Particle Swarm Optimization, Extremum Seeking, Perturbation, Cooperative Search

1 Introduction

Swarm robotics is an area that has received a lot of attention from worldwide researchers. Using a mobile robot swarm to search targets is a typical topic in this area. Swarm robotic systems usually consist of many identical or similar simple individuals but can give super behavior in swarms. However, a swarm robotic system not only includes multiple robots but also the swarm intelligence from collaboration between the members. The methods used for controlling swarm robotics mainly boil down to two categories. The traditional ones like, e.g., artificial potential fields, or exact cell decomposition, are just inadequate when performing complex tasks. Another kind of methods is referred as non-traditional, like bacterial colony algorithms, reactive immune network, Particle Swarm Optimization (PSO), and Extremum Seeking (ES). Among them the PSO and ES are especially appealing due to their unique features, see [4], [7].

The PSO is originally only used as an optimization method, although it is extended and utilized in the robotics area, see example in [6]. The work [6] extends PSO to mechanical PSO which takes into account the mechanical properties of real robots, see also Section 2.2. Together with other strategies, encouraging results are obtained by this method. However, it requires relatively precise localization for forming the feedback loop and it is difficult to realize fast online driving due to sensor delays. Another side, Extremum Seeking is applicable as a means of navigating robots in environments where robot positions are unavailable [7]. However, the basic ES is non-cooperative, i.e., each robot is driven by ES individually. Motivated by this, this article investigates the swarm robots cooperative search by integrating mechanical PSO and ES.

2 Algorithm Design

2.1 Investigation Prerequisites

Before designing the algorithm for swarm robots, some prerequisites have to be stated. The robots used in the swarm are mobile robots. They are assumed to be relatively simple and do not have the capability of localizing their own positions. However, they are capable of sensing the relative states of their neighbors and the signal strength from the target. The simulated robots are here considered as 2D mass points without volumes.

Secondly, there are no obstacles included at the moment in the environment. The spatial distribution of the searched signal originating from the target is unknown to the robots, neither the position of the target. However, in this study the target is sending a signal which is known to decay with the distance from the source. As a usual source distribution we use the quadratic form

$$f(x, y) = f^* - q_x(x - x^*)^2 - q_y(y - y^*)^2 \tag{1}$$

to describe it. Here f is the detected signal strength, (x^*, y^*) is the maximizer while f^* represents the maximum, q_x and q_y are positive constants.

2.2 From Basic PSO to Mechanical PSO

The PSO was inspired from some biological populations, for instance, the swarm of birds. Each bird is taken as an adaptive agent and can communicate with the environment and other agents. During the process of communication, they will 'learn' or 'accumulate experience' in order to change the structure and behavior of the swarm. Such processes are the basis of PSO. In PSO, the size of the swarm is denoted by N_p and the members are called particles. The 'velocity' and position of particles are represented by $N_p \times n$ matrices $\dot{\boldsymbol{x}}$ and $\boldsymbol{x}$, respectively.

The recursion of one commonly used form of basic PSO for all N_p particles is

$$\begin{bmatrix} \boldsymbol{x}^{s+1} \\ \dot{\boldsymbol{x}}^{s+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}^{s} \\ \omega_p \dot{\boldsymbol{x}}^{s} \end{bmatrix} + \begin{bmatrix} \dot{\boldsymbol{x}}^{s+1} \\ c_1 \boldsymbol{r}_1^s \cdot (\boldsymbol{x}_{self}^{best,s} - \boldsymbol{x}^s) + c_2 \boldsymbol{r}_2^s \cdot (\hat{\boldsymbol{x}}_{swarm}^{best,s} - \boldsymbol{x}^s) \end{bmatrix} . \tag{2}$$

Here s denotes the iterative steps. The right-hand side of the second line of (2) contains three components, i.e., the 'inertia' which models the particles tendency of last step; the 'memory' which means moving towards the particles' self best positions, respectively; and the 'cooperation' which drives the particles to the swarm best position. In (2), c_1 and c_2 are usually non-negative constant real numbers while random effects are kept in $\boldsymbol{r}_1^s$ and $\boldsymbol{r}_2^s$. Detailed definitions can be found in [6].

We consider one particle to represent one robot since the particles in PSO looking for the minimum (or maximum) of an objective function according to their update formulae is quite similar to the robots search scenario in which the robots are searching a target according to their cooperatively generated trajectories. Many of their correspondences are summarized in [6]. We interpret the PSO-based algorithm as providing the required forces in the view of multibody system dynamics. Namely, each robot is considered as one body in a multibody system which is influenced by forces and torques but without direct mechanical connections. In addition, the particles are replaced by mechanical robots whose motions follow physical laws. This is done in order to generate physically reasonable search trajectories. For considering the feasible dynamics, the inertia, and other physical features of the robots, the basic PSO algorithm is extended.

In a general way, if one defines $\boldsymbol{k}$ coming from Euler equations, and $\boldsymbol{q}$ contains the information of external forces and torques acting on all N_p robots, the acceleration of the entire robot swarm can be formulated by

$$\ddot{\boldsymbol{x}} = \left[\ddot{\boldsymbol{x}}_1 \; \ddot{\boldsymbol{x}}_2 \; \cdots \; \ddot{\boldsymbol{x}}_i \; \cdots \; \ddot{\boldsymbol{x}}_{N_p}\right]^T = \boldsymbol{M}^{-1} \cdot (\boldsymbol{q} - \boldsymbol{k}) = \boldsymbol{M}^{-1} \cdot \boldsymbol{F} \qquad \in \mathbb{R}^{3N_p \times 1}. \tag{3}$$

With the state vector $\boldsymbol{y}_{st} = \left[\boldsymbol{x} \; \dot{\boldsymbol{x}}\right]^T$, state equation

$$\dot{\boldsymbol{y}}_{st} = \begin{bmatrix} \dot{\boldsymbol{x}} \\ \boldsymbol{M}^{-1} \cdot \boldsymbol{F} \end{bmatrix}, \quad \text{and Euler forward integration} \quad \boldsymbol{y}_{st}^{s+1} = \boldsymbol{y}_{st}^{s} + \Delta t \, \dot{\boldsymbol{y}}_{st}^{s}, \quad \text{it yields} \tag{4}$$

$$\begin{bmatrix} \boldsymbol{x}^{s+1} \\ \dot{\boldsymbol{x}}^{s+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}^{s} \\ \dot{\boldsymbol{x}}^{s} \end{bmatrix} + \Delta t \begin{bmatrix} \dot{\boldsymbol{x}}^{s} \\ \boldsymbol{M}^{-1} \cdot \boldsymbol{F}^{s} \end{bmatrix}. \tag{5}$$

We define the robot to be only influenced by forces, i.e., $l_i = 0$ at the moment. The force $\boldsymbol{F}^s$ is further determined by three parts, $\boldsymbol{f}_1^s$, $\boldsymbol{f}_2^s$ and $\boldsymbol{f}_3^s$, which are

$$\boldsymbol{f}_1^s = -\boldsymbol{h}_{f_1}^s \cdot \left(\boldsymbol{x}^s - \boldsymbol{x}_{self}^{best,s}\right), \;\; \boldsymbol{f}_2^s = -\boldsymbol{h}_{f_2}^s \cdot \left(\boldsymbol{x}^s - \hat{\boldsymbol{x}}_{swarm}^{best,s}\right), \;\; \boldsymbol{f}_3^s = -\boldsymbol{h}_{f_3}^s \cdot \dot{\boldsymbol{x}}^s. \tag{6}$$

Here $\boldsymbol{f}_1^s$, $\boldsymbol{f}_2^s$ and $\boldsymbol{f}_3^s$ contain physical meanings corresponding to the ‘memory’, ‘cooperation’, and ‘inertia’ phenomena in basic PSO. Combining (5) and (6) yields

$$\begin{aligned} \begin{bmatrix} \boldsymbol{x}^{s+1} \\ \dot{\boldsymbol{x}}^{s+1} \end{bmatrix} &= \begin{bmatrix} \boldsymbol{x}^{s} \\ \left(\boldsymbol{I}_{3N_p} - \Delta t \, \boldsymbol{M}^{-1} \cdot \boldsymbol{h}_{f_3}^s\right) \cdot \dot{\boldsymbol{x}}^s \end{bmatrix} \\ &\quad + \Delta t \begin{bmatrix} \dot{\boldsymbol{x}}^s \\ \boldsymbol{M}^{-1} \cdot \boldsymbol{h}_{f_1}^s \cdot \left(\boldsymbol{x}_{self}^{best,s} - \boldsymbol{x}^s\right) + \boldsymbol{M}^{-1} \cdot \boldsymbol{h}_{f_2}^s \cdot \left(\hat{\boldsymbol{x}}_{swarm}^{best,s} - \boldsymbol{x}^s\right) \end{bmatrix}. \end{aligned} \tag{7}$$

For more detailed derivations and explanations please refer to [6]. Equation (7) is the developed mechanical PSO which is used to generate cooperatively physically reasonable trajectories for swarm mobile robots. However, it requires heavy robot localization.

2.3 Perturbation Based Extremum Seeking

Extremum Seeking (ES) had been proven to be a powerful tool in real-time non-model based control and optimization [1]. Recently, ES also has been used for swarm networked agents with each member only sensing limited local information [5]. Extremum Seeking usually is applied for questions as for seeking the maxima of objective functions. Due to its non-model based character, it is applicable to control problems which contain nonlinearity either in the plant or in its control objective. For example, in the case of cooperative search performed by swarm robots, either the robot models or the distribution of the source, or both of them can be nonlinear since ES based methods are possible to use only one external signal whose strength is detected by robots and the specific distribution form of the signal is not critical. It doesn't care about the actual positions of robots. Furthermore, the linearity or nonlinearity of the robot model is not important. Nonetheless, the signal strength f is only a single dimensional information. It is not directly sufficient for guiding robots since it lacks the 'gradient' information of the target signal. One way to solve this issue is that of equipping gradient detecting sensors on robots. However, this is a big challenge and we prefer that the robot only needs to detect one external signal. The amazing thing happens when the basic ES is varied by perturbation which is hardware free and easy for implementation. Relying on its persistence of excitation, usually a sinusoidal signal, the perturbation based ES perturbs the parameters being tuned. Through this method, the gradient information is obtained. We use the perturbation based ES scheme similarly as in [8] for guiding robots. Its control block diagram for a single robot is shown in Fig. 1.

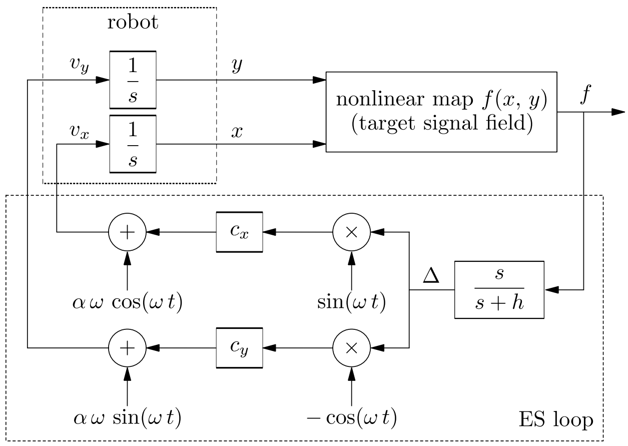


Fig. 1. Extremum Seeking scheme with x, y axes velocities as inputs for robot

In Fig. 1 the parameters α, ω, c_x, c_y and h are chosen by the designers. The washout filter $s/(s+h)$ filters out the DC component of f, then the two-channel perturbations generate gradient estimates of f in x and y directions which usually are unmeasurable by physical sensors. After the ES loop, the velocity inputs are tuned for driving the robot. The used control laws are then governed by

$$v_x = \alpha\,\omega\,\cos(\omega t) + c_x \Delta \sin(\omega t),\; v_y = \alpha\,\omega\,\sin(\omega t) - c_y \Delta \cos(\omega t),\; \Delta = \frac{s}{s+h}[f]. \quad (8)$$

The control laws in (8) are actually optional. Dürr et al. [2] used different perturbations which also work well. Unfortunately, due to the sinusoidal perturbation, the trajectory generated from the ES based method is spiral like which artificially increases the travel distance. This is extremely serious when the distance between start position and target position is far, because the uncertainty from the AC part of f is increased. This is not acceptable for robot practical implementation. Furthermore, if this ES scheme is used for swarm robots, all robots are non-cooperative. Therefore, we try to integrate it into mechanical PSO while the localization free feature is inherited.

2.4 Mechanical Particle Swarm Optimization Aided by Extremum Seeking

Our purpose is to integrate the cooperation benefits of mechanical PSO and the localization free feature of ES. The ES is aiding the overall search algorithm of mechanical PSO. The mechanical PSO is used to generate intermediate states for guiding robots. From each state to its next adjacent state there is only a short distance. The perturbation based ES only needs to drive the robot to the next state while temporary taking the next state as its current target, see Fig. 2 for the relationship of mechanical PSO and ES.

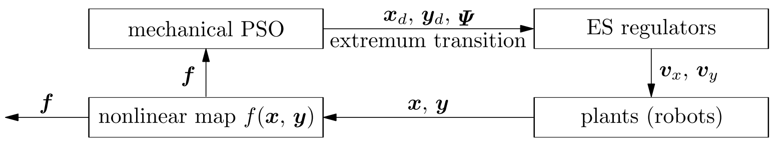


Fig. 2. Extremum Seeking aids for mechanical PSO

One should not forget that the intermediate states are not target positions. Thus, the perturbation based ES doesn't directly qualify. So, the maximum transition must be performed through which the target source (with maximal signal strength) is mathematically and temporary transited to the desired state by using information $\boldsymbol{f}$ from the actual source and $\boldsymbol{\Psi}$ from robots relative observation. This idea is also expressed by

$$\boldsymbol{f}^* = \boldsymbol{f}(\boldsymbol{x}^*, \boldsymbol{y}^*) \quad \xrightarrow[\boldsymbol{g}()]{\boldsymbol{f}, \boldsymbol{\Psi}} \quad \boldsymbol{g}^* = \boldsymbol{g}(\boldsymbol{x}_d, \boldsymbol{y}_d, \boldsymbol{f}, \boldsymbol{\Psi}) \tag{9}$$

where $\boldsymbol{f}$ is a vector containing all the robots detected signal strengths. To be emphasized, $\boldsymbol{f}$ is from the actual source and it is the only signal that the robots can detect since the intermediate 'targets' $(\boldsymbol{x}_d, \boldsymbol{y}_d)$ from mechanical PSO are artificial. In (9) $\boldsymbol{g}^*$ are the new maxima, $\boldsymbol{g}$ is a function which corresponds to the function $\boldsymbol{f}$ and describes the artificial targets at the intermediate states. In this study, we assume there is such a function $\boldsymbol{g}$. By this way, each robot senses the current 'target' which locates at the corresponding state from a step of mechanical PSO. Then, our perturbation based ES drives the robot to the desired state with relatively stable trajectory due to the short distance between two adjacent states. Looking at the whole robot swarm, they are still moving in a cooperative manner since each robot traces one trajectory from mechanical PSO.

The procedures of the whole method is summarized in Algorithm 1. The method used in this study is very different to the work in [4] where PSO is still used in the view of optimization without considering the robots physical properties. The PSO generated states in [4] require a re-generation to smooth the trajectory, whereas the states from mechanical PSO in this study are ready to be traced with physically reasonable quality. Thus, the PSO in [4] has no obvious advantages compared to some other swarm intelligence based algorithms like, e.g., ant colony, bacteria foraging. In addition, [4] only handles a single robot and the particles are virtually without mapping to real robots. Some other researches, e.g., [3] and [7], provide variants of ES for swarm seeking which are basically formation control oriented. Furthermore, their control cost and energy consumption from the not well organized trajectories have restricted their applications although they are also cooperative. In contrast, the scheme in this study is more straightforward and feasible considering the implementation.

Algorithm 1 Mechanical PSO aided by ES for robots cooperative search

1: /* initialize: give all required control parameters, read in start positions and initial signal strength $\boldsymbol{f}^0$ of all robots, mechanical PSO step $s = 0$, define stop criteria */
2: update mechanical PSO using (7), provides $(\boldsymbol{x}_d, \boldsymbol{y}_d)$, $s = s + 1$, robot index $i = 1$
3: obtain $\boldsymbol{\Psi}_i^s$ by relative observation, perform maximum transition using (9)
4: perturbation based ES regulates robot i to $(x_{i,d}, y_{i,d})$ due to (8), $i = i + 1$
5: repeat steps 3-4 until $i > N_p$
6: measure $\boldsymbol{f}^s$ at new positions, evaluate new $\boldsymbol{f}^s$ for mechanical PSO
7: repeat steps 2-6 until a stop criterion is met

3 Simulation

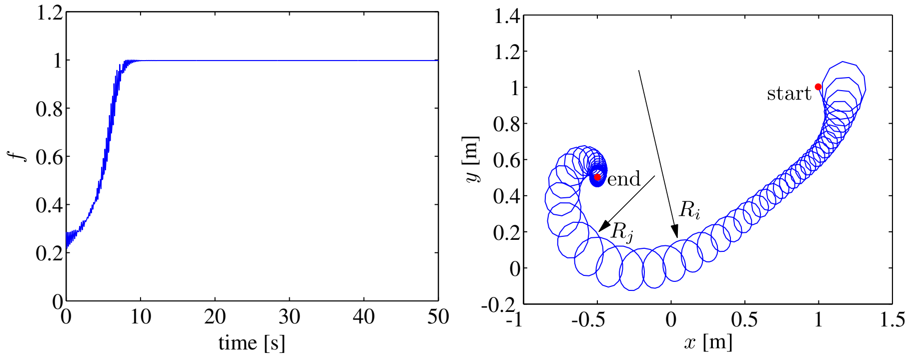
3.1 Simulation Setup

In our simulations, the robots are assumed to run in a $3\,\mathrm{m} \times 3\,\mathrm{m}$ environment. The used parameters for mechanical PSO are $\omega_p = 0.6$, $c_1 = 0.1$, $c_2 = 0.8$, for ES are $\omega = 100, 125, \ldots$, (different for each robot) $\alpha = 0.05$, $c_x = c_y = 10$. The weights of the target field function are $q_x = q_y = 1$. We set in simulation the final maximizer at $(x^*, y^*) = (-0.5, 0.5)$. During the search the maximizers are the corresponding states generated from mechanical PSO, the final target maximum is set to $f^* = 1$.

3.2 Swarm Robots Cooperative Search by Mechanical PSO Aided by ES

We first verify a single robot to be driven by our perturbation based ES controller. The robot is actuated by x and y axes velocities and is supposed to move from $(1, 1)$ directly to $(-0.5, 0.5)$ without intermediate stops. Figure 3(a) shows the performance measured by this robot, from which one can observe the change of the detected signal. After about 10 s (simulation time), it approaches the maximum which means the robot is very close to the target. Figure 3(b) demonstrates the robot trajectory which looks like a spiral curve. In Fig. 3(b) the marked R_i, R_j are the revolution radii of the robot motion from which one can see its significant change. The rotation radii of the robot motion are changing, too. In addition, both of them are unpredictable. This kind of changes will become more intense with increasing distance to the target.

After this we now set up a four robots search scenario which integrates mechanical PSO and perturbation based ES. The robot trajectories are shown in Fig. 4. In Fig. 4



(a) Performance output (b) Robot trajectory

Fig. 3. Extremum Seeking moves one robot from $(1, 1)$ to $(-0.5, 0.5)$

the green, red, pink and black dots (lines) are the mechanical PSO computed states (trajectories). The distances between two adjacent states are smaller. The overlapping spiral like blue curves are the ES regulated trajectories which have relatively stable spiral radii. This is very helpful for real robots implementation. From Fig. 4 one can see that the method of mechanical PSO aided by ES is feasible. Importantly, through the whole process, no robot localization is required. The blue trajectories are still longer than the ones obtained by directly connecting mechanical PSO states. This is negative but in exchange there is no localization required. From a macro point of view, the robots are still traveling cooperatively. The mechanical PSO guides the robots not to move too arbitrary as when only driven by ES while on another side the perturbation based ES frees the robots from localization.

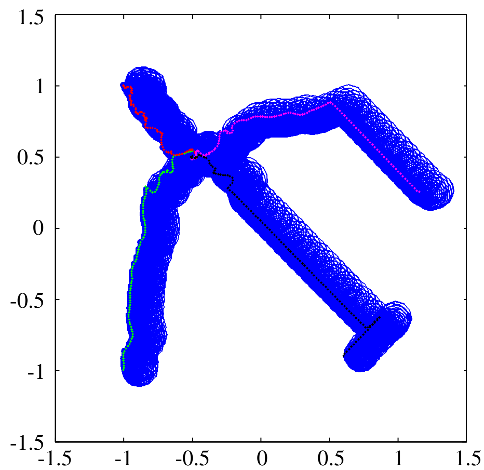


Fig. 4. Cooperative search trajectories (mechanical PSO aided by ES)

4 Open Questions and Discussion

The method gets rid of robot localization which gives the implementation a lot of freedom. However, the energy consumed from the ES trajectory is much higher than the one from mechanical PSO. How to adjust the parameters of the ES controller for energy saving is an interesting consequent investigation point.

If robot volume and obstacles are included, algorithm improvement is highly demanded. Furthermore, the maximum transition of (9) actually is a very strong assumption. Performing relative observation and building the function $\boldsymbol{g}$ are not easy when considering the real robots implementations.

5 Conclusion

For the swarm mobile robots cooperative search, this investigation has integrated advantages both from mechanical Particle Swarm Optimization and Extremum Seeking. The mechanical PSO provides cooperative search trajectories based on the consideration of real robots, while perturbation based ES is responsible for regulating the robots towards the purposed states from mechanical PSO. This method no longer needs the localization of the moving robots which is usually required by traditional robot navigation. This will probably open a new research window for swarm mobile robots cooperative search. The feasibility of the conducted method in this pilot study is investigated by simulation.

Acknowledgment. The authors would like to thank Dipl.-Ing. Hans-Bernd Dürr, IST, University of Stuttgart, for his valuable suggestions and helpful discussions on Extremum Seeking control.

References

1. Ariyur, K.B., Krstić, M.: Real-Time Optimization by Extremum-Seeking Control. Wiley-Interscience, Hoboken (2003)
2. Dürr, H.B., Stanković, M.S., Ebenbauer, C., Johansson, K.H.: Lie bracket approximation of extremum seeking systems. Automatica (2012), (accepted for publication)
3. Ghods, N., Krstić, M.: Multiagent deployment over a source. IEEE Transactions on Control Systems Technology 20(1), 277–285 (2012)
4. Hong, C., Li, K.: Swarm intelligence-based extremum seeking control. Expert Systems with Applications 38(12), 14852–14860 (2011)
5. Stanković, M.S., Johansson, K.H., Stipanović, D.M.: Distributed seeking of Nash equilibria with applications to mobile sensor networks. IEEE Transactions on Automatic Control 57(4), 904–919 (2012)
6. Tang, Q.: Cooperative Search by Mixed Simulated and Real Robots in a Swarm Based on Mechanical Particle Swarm Optimization. Doctoral thesis, University of Stuttgart, No. 25. Aachen: Shaker Verlag (2012)
7. Zhang, C., Ordóñez, R.: Extremum-Seeking Control and Applications: A Numerical Optimization-Based Approach. Springer, London (2011)
8. Zhang, C., Siranosian, A., Krstić, M.: Extremum seeking for moderately unstable systems and for autonomous vehicle target tracking without position measurements. Automatica 43(10), 1832–1839 (2007)