**Proceedings of ASME 2016 International Design Engineering Technical Conferences and
Computers and Information in Engineering Conference
IDETC2016
August 21-24, 2016, Charlotte, North Carolina**

IDETC2016-59984

AN APPROACH TO SORTING SWARM ROBOTS TO OPTIMIZE PERFORMANCE.

Beining Shang, Richard M. Crowder[*]**, Klaus-Peter Zauner**

Department of Electronics and Computer Science
University of Southampton
Southampton, Hampshire, SO15 1BJ
UK
rmc@ecs.soton.ac.uk

ABSTRACT

Swarm robotic systems can offer many advantages including robustness, flexibility and scalability. However one of the issues relating to overall swarm performance that needs to be considered is hardware variations inherent in the implementation of individual swarm robots. This variation can bring behavioral diversity within the swarm, resulting in uncontrollable swarm behaviors, low efficiency, etc. If swarm robots could be separated by behaviors, operational advantages could be obtained. In this paper we report an approach to the sorting of large robotic swarms using an approach inspired by chromatography. Hence the tedious and expensive calibration process can be avoided. The results investigate the influence of the internal control parameters, together with environmental effects on the robotic behavioral sorting. We concluded that if the robot has knowledge of previous events coupled with a specific arena pattern density will offer improved behavioral sorting.

1 INTRODUCTION

Hardware-based swarm robotic research is faced with issues resulting from variations in the hardware. These variations result from components manufacture, assembly variations and degradation over use [1–3], as summarized in Figure 1. A number of calibration techniques have been discussed in the literature to tackle this problem [4, 5], however robotic calibration is only cost-effective on large and expensive platforms such as industrial manipulators and not small mass-produced swam robots, such as the University of Southampton's Formica robot [6]. Although individual robots in the swarm are usually simple, it still takes time and efforts to calibrate, considering that a robotic swarm usually consists of a large number of individuals. In the context of swarm robots, hardware variation problem have been demonstrated to influence the robot's behaviors and change the results of the experiments due to the non-linear interactions between swarm robots and environment [7]. As previously reported, an approach has been proposed in [8] to sort robotic swarm members according to their behaviors by adopting a technique inspired by chromatography, where the separation of a mixture of compounds is achieved by each compound taking different times to travel through a stationary medium.

The sorting of robotic swarms will provide improved swarm efficiency by allowing the deployment of swarm robots with the required behaviors rather than directly using a large number of robots with unfavored behaviors caused by their hardware variations.

In this paper we simulate the separation of small line-following robots within an arena that consists of short reflective lines of random orientation and differing densities. In Section 2 we define the research problem. In Section 3 we discuss the model of the robot and its control architecture. In this paper we compare the performance of robots with hardware variation as a function of the robots memory length and arena density. The

---

[*]Address all correspondence to this author.
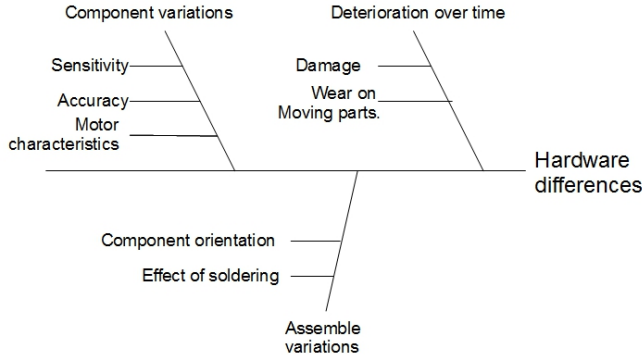
FIGURE 1. Overview of possible reasons for hardware difference within swarm robots built to an identical design.

length of the robot's memory vector is equivalent to the integral term in a conventional PID controller, allowing the robot's previous interactions with the environment to influence the current behavior. The separation technique is discussed in Section 4. The results are presented and discussed in Section 5, with the conclusions in Section 6.

2 PROBLEM DESCRIPTION

Elliott and Shadbolt [3] argued that swarm robots are heterogeneous at hardware level even if the robots are manufactured to the same design and specification. They further argued that although it is very difficult and not cost-effective, identical behaviors of hardware-based robots in a swarm still can be made, in which particular software needs to be fine-tuned to compensate for the inherent hardware differences. In practice, tuning the software controller for individual robots in a swarm is not a feasible option.

Efforts have been made to develop software which is able to withstand individual hardware differences to achieve similar behaviors among swarm robots. While studying the impact of sensor offsets and scaling factors to swarm robotic learning capabilities through simulation. Pugh and Martinoli [9] found that in the case of evolving obstacle avoidance, both genetic algorithms and particle swarm optimization allowed the robots to withstand small variations in sensor offsets and large variations in sensor scaling factors, however the robots showed poor performance with high offset variations.

In summary, because of the highly non-linearity of swarm robot controller, hardware variations will be amplified and force swarm-based heterogeneous behaviors.

An individual robot has its unique hardware specification by which its information perception and reactions are influenced. Thus every robot has its own behavioral characteristics. As the magnitudes of hardware differences are typically very small, the characteristics of robots on the behavior level are almost unde-

tectable under most conditions. However by encouraging numerous interactions between robots and environment, these tiny behavioral differences accumulate and can then be perceived on a higher level such as robots' coordinates after the completion of a specific experiments. Therefore the more interactions, the more behavioral diversity can be traced.

In addition, if the robot has the ability to remember its previous experience, it will continuously react in a particular fashion until the memory of of the previous event fades away. Hence the robot's reactions to its previous experiences will be modified the robot's behavioral characteristics. Hence, the robot's memory should be able to intensify the effect of interactions between the robot and the environment. Thus it is assumed that with memory ability, better separation of robots can be obtained during the swarm robotic behavioral sorting experiments.

If a line-following robot is operating at a predefined arena, the arena pattern density is also related with the number of interactions between the robot and environment. If the pattern density is high, the chance of robot's encountering a line is higher, and more interactions can happen. And better separation of robot's behaviors should be obtained. Thus it is also assumed that better separation of robots can be obtained with high arena pattern densities.

Differences between any two seemingly identical robots can emerge both when they are manufactured and then when used. Variations result from differences in the manufacture procedures including components specification and assembly [10]. For example, the same type of sensors used on any two robots can have marginally different sensitivity [5]. In addition actuators can have different output profiles and batteries have different characteristics under load. Once the robots have been assembled, the positions of components and soldering parameters vary. For instance, the orientation of IR sensors [11] or wheels will be subject to variation. Finally when in use, different robots experience different circumstances of wear and tear or even damage, such as sensor aging, decreasing battery capacity, and mechanical deterioration. This is especially the case in swarms where the material and calibration cost per unit has to be very low.

3 THE LINE-FOLLOWING ROBOT

The swarm robot used in the research reported in this paper is based on a conventional line-following design often used in robotic swarm scenarios. As shown in Fig 2, the robot is equipped with two wheels driven by individual motors, allowing differential steering. The robot is able to following a highly reflective track, through the use of two infra-red photoelectric sensors. It should be noted that only the essential parts of the robot's dynamics are considered, for example, the rolling friction of the wheels and stiction within the drive train not modeled. Although modeling every detailed aspect of the system as such only involves a further set of parameters, it would add one more layer

of complexity to the problem, which would make the drawing of conclusion more difficult. In addition it would be computationally more expensive.

The key points of the design are briefly discussed in the following subsections.
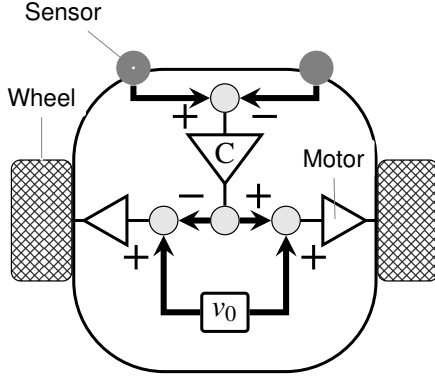
FIGURE 2. A diagrammatic representation of the robot and its controller. Two infra-red sensors in located at the front of the robot, the voltage difference between their outputs is amplified by the proportional, integral and differential controller denoted by C. A constant voltage, V_0, is added to the output of the controller to ensures that the robot moves forward if the outputs from the two sensors are equal.

3.1 Driving Train

A conventional differential steering approach was used, where the two driven wheels are typically powered by brushed D.C. (direct current) motors, as shown in Fig 2. The drive train are modeled as pure gain, m_L and m_R, to simplify the modeling of the individual power amplifier, motor and associated gearbox. In the control scheme used, the voltage, V_o, determines the forward speed. V_{out} is the voltage difference between output of two infra-red sensors.

$$\omega_L(t) = (V_o + V_{out}(t)) \, m_L$$
$$\omega_R(t) = (V_o - V_{out}(t)) \, m_R \tag{1}$$

Based on the speeds of the individual motors, the effective linear velocity of the robot can determined using the radius of the individual wheels r_L and r_R, hence the robot's linear ($\dot{x}$ and $\dot{y}$) and turning $\dot{\phi}$ velocities can be calculated:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} -\frac{r_L \sin\phi}{2} & -\frac{r_R \sin\phi}{2} \\ \frac{r_L \cos\phi}{2} & \frac{r_L \cos\phi}{2} \\ -\frac{r_L}{b} & \frac{r_R}{b} \end{bmatrix} \begin{bmatrix} \omega_L(t) \\ \omega_R(t) \end{bmatrix} \tag{2}$$

where b is the distance between the wheels.

3.2 Sensors

The two downward pointing infra-red sensors are located at the front of the robot. The sensor's response [12] to the reflective line is modeled using Equation 3,

$$S_T = \sum_{1}^{n} \left[\frac{\alpha}{x_n^2} \cos\theta_n + \beta \right] \text{ for } \theta_n \leq \measuredangle v \tag{3}$$

As the reflective line can be considered as a structure of multiple consecutive points, reflecting light to the sensor, the sensor's output for a reflective line can be regarded as the sensor's response to these multiple points located within the sensor's viewing angle, $\measuredangle v$. The sensor output to a single reflecting point is a function of the incidence angle θ of the reflective light and the distance x between the sensor and the reflecting point, while α determines the sensitivity of the sensor, and β models the sensor's output offset and the effect of ambient light.

3.3 Controller

In the line-following scenario used in this paper, the robot's controller will try to keep the output of left and right infra-red sensors identical, if not the robot will be driven to change its relative position to the line to correct any in-balance. The controller is based on a conventional PID design. In a PID controller the integral term, I, is proportional to both the magnitude of the error and the duration of the error and can be considered to be a summation over a period of time. In our approach the integral term is simulated by the summation along a vector that stores the robots previous experience at discrete time intervals, which in the line following scenario is the difference between the outputs of two line following sensors.

It is assumed that the experience gained by the robot at each time steps should have a different impact on the swarm robot's behaviors, therefore the memory value's for each time step needs to be assigned with an individual weight before summation, as shown in Equation 5.

$$I_n = w_n \delta_{t_n} \tag{4}$$

where w_n is the weight and δ_{t_n} is the sensor difference value at time step t_n. As the memory weighting parameter is a vector with the same length of the robot's memory, it is not realistic to train individual elements of the vector which can be up to 900 terms in length, therefore a sinusoid-shape curve is used to represent the memory weighting vector. The weighting sinusoid is defined

by its amplitude, A, frequency, a, phase shift, ϕ, and offset, Δ, as shown in Equation 6.

$$w(n) = \frac{[A \, \sin(an+\phi)+\Delta]}{\sum_{n=1}^{ml}[A \, \sin(an+\phi)+\Delta]} \qquad (5)$$

The controller C shown in Fig. 2 takes the difference between the two sensors' output and determines the required output for the two motors. The voltage offset V_0 ensures that the swarm robot is biased to move forward. The implementation is given in Equation 7.

$$
\begin{aligned}
V_{out}(t) &= p \, \delta_t + i \sum_{0}^{n=ml} I_n + d \, (\delta_t - \delta_{t_1}) \\
V_{Lm}(t) &= V_0 - V_{out}(t) \\
V_{Rm}(t) &= V_0 + V_{out}(t)
\end{aligned}
\qquad (6)
$$

where δ_t is difference between the outputs of the two sensors t, at the current time step, δ_{t_1} is the difference for the previous time step. The parameters p, i and d are the proportional, integral and differential gains respectively, and ml is the number of memory steps available in the controller. I_n is the value stored in the memory which has been weighted and summed for the integral parts in the PID controller. The resultant voltage V_{out} is then offset by V_0 to generate the voltages V_{Lm} and V_{Rm} which drive the left and right motors.

4 METHODOLOGY

In this paper we consider several groups of typical line-following swarm robots. The robots in a specific group have variations in their hardware parameters, but identical memory lengths.

4.1 Hardware Variation

Real robots within a swarm come with unavoidable hardware variations [7]. These variations can normally be found in the swarm robots' hardware parameters, such as sensor viewing angle, motor gain etc. Normally the values of the hardware parameters follow Gaussian distributions. To model the hardware variations in the context of robot swarm, the standard swarm robot discussed above is firstly generated, from which other members in the robot swarm are derived with varied values for hardware parameters which follow Gaussian distribution.

4.2 Robot sorting

We implemented a separation process for robots that was inspired by planar chromatography. Planar chromatography is a separation technique in which the stationary phase is present as or on a plane such as paper. The different compounds in the sample mixture travel different distances according to how strongly they interact with the stationary phase as compared to the mobile phase giving the characteristic bands [13, 14]. In the approach used in this paper, swarm robots (equivalent to different compounds in a solvent) are required to travel across an arena from left to right (this is equivalent to the stationary phase) hence achieving separation due to the different iteration with the arena.

In the simulation all the swarm robots start at the same location in the arena, with identical orientation. To replicate the interaction between the phases as found in chromatography, a complex pattern is applied to the arena as discussed in Section 4.3. Finally we apply a *quasi-wind* which was designed to push all the swarm robots to the right, across the arena. In chromatography this force is usually gravity. The effect of the *quasi-wind* applied to individual swarm robot is expressed in Equation 8.

$$
\begin{aligned}
x_j^t &= x_j^{t_1} + F_b \cdot \left|\sin(ori_j^{t_1})\right| \\
ori_j^t &= \begin{cases}
ori_j^{t_1} + F_a \cdot \left|\tan(ori_j^{t_1})\right| & \text{if } \sin(ori_j^{t_1}) > 0 \\
ori_j^{t_1} - F_a \cdot \left|\tan(ori_j^{t_1})\right| & \text{if } \sin(ori_j^{t_1}) < 0 \\
ori_j^{t_1} & \text{if } \sin(ori_j^{t_1}) = 0
\end{cases}
\end{aligned}
\qquad (7)
$$

where x_j^t is the X coordinate of swarm robot j at simulation time t and $x_j^{t_1}$ is the X coordinate of swarm robot j at simulation time $t-1$; ori_j^t is the orientation of swarm robot j at simulation time t; F_a is the force of the *quasi-wind* affecting the orientation of the swarm robot and F_b is the force of the *quasi-wind* influencing the swarm robot's X coordinate. The equation can be interpreted as if the orientation of the swarm robot and direction of the *quasi-wind* is different, the speed of each swarm robot at each simulation step is the vector addition of the swarm robot's own speed and a subcomponent of speed which is due to the *quasi-wind*, and then an angular speed is added to the swarm robot which force the swarm robot to rotate to the right.

4.3 Arenas used for simulation

To determine the relationship between swarm robot's memory length and the environment conditions, nine arenas with different pattern densities were generated, as shown in Figure 4. In each of the arenas, the patterns consist of short lines with random orientation which are located on a predefined grid. The density of the arena pattern is defined by the ratio of the pattern line length to the distance between the centers of two adjacent patterns, Figure 3. In total nine arenas with densities of 0.1, 0.2, ..., 0.9 were used in the simulations.

As discussed in [8], the separation of the swarm robots is
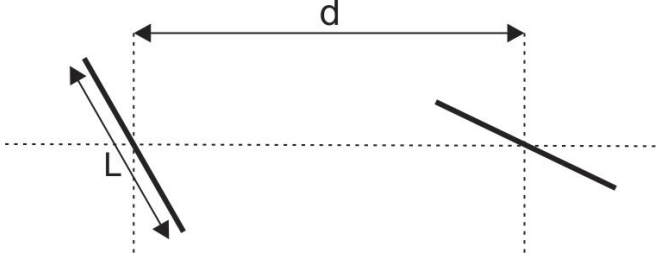
FIGURE 3. The basic elements of the arena pattern. Each line is of a fixed length L and is randomly orientated. The arena pattern density is determined by the distance d, giving an overall density measure of $\frac{L}{d}$.

dependent on the large number of interactions. Therefore all arenas are effectively considered to be infinitely long, allowing the maximum possible interactions between swarm robots and arena patterns to be achieved within the run time of simulation. In addition, to reduce the complexity of the problems, it is assumed that when a swarm robot reaches either the upper or lower boundary of the arena, the swarm robot will re-appear on the other boundary with the identical orientation. For this configuration the separation of the robots can be described using a single parameter, the X coordinate of the robot's final position.

(a) *Density* = 0.2

(b) *Density* = 0.4

(c) *Density* = 0.6

(d) *Density* = 0.8
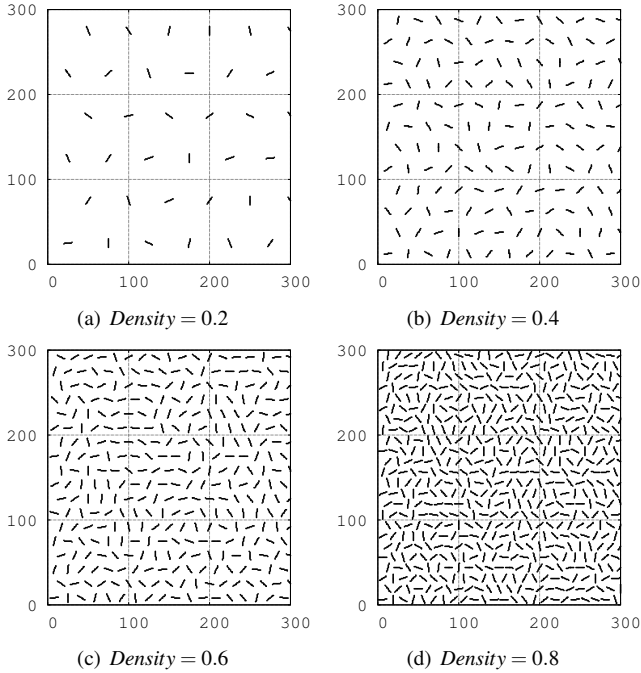
FIGURE 4. Testing Arenas with different pattern densities. Only a 300×300 portion of arenas is shown.

4.4 Swarm Robot Training

The training of each individual swarm robot involves of the determination of several controller parameters including the proportional, integral, differential coefficients p, i, d of the PID controller and the memory weighting parameter w_n to make the swarm robot achieve optimal performance, Table 1.

TABLE 1.   Controller Parameters to be Determined during Training

Parameter	Description	Description
p	Proportional gain	
i	Integral gain	PID controller
d	Differential gain	
A	Amplitude	
a	Frequency	Sinusoid used for memory weighting vector w_n
ϕ	Phase shift	
Δ	Offset	

There are two reasons for selecting a sinusoid-shape curve to generate the memory weighting vector w_n. Firstly, the sine wave is continuous, the memory of a robot is also continuous and it is rare that a particular part of the memory will dramatically influence the overall behavior of the swarm robotic and secondly, with the four parameters given in Table 1, the sine wave can have a variety of forms, if necessary, it can emphasize or make little impact on certain parts of the swarm robot's memory.

A Sobol sequence was used to optimize the seven parameters given in Table 1, as the numbers generated from a Sobol sequence give uniform distribution over the search space and the optimal parameter values can be found with fewer number of runs [15].

During each training run, the swarm robot was required to follow a single period sinusoid training line, as the swarm robot's ability of both turning left and right would be trained without any bias. For each robot in the swarm, the task will be repeated for a number of times to test different sets of controller parameters. To evaluate the training results, the swarm robot's trajectory generated has to be compared with the training line. The set of parameters which makes the swarm robot achieve the best following results between the training line and the resultant trajectory were selected to be the optimal controller parameters for that particular swarm robot.

Due to the lengthy time taken to run the training trials, the University of Southampton's Iridis[1] Compute Cluster was used

---

[1] http://www.southampton.ac.uk/isolutions/staff/high-performance-computing.page

for this research.

5 SIMULATION RESULTS AND DISCUSSION

In order to investigate the influence of swarm robotic memory length and arena pattern density to the separation of the swarm robots, we ran the simulations with swarm robots which have different memory lengths in arenas with different pattern densities. The simulations were configured as follows:

1. Nine arenas are generated and their densities are 0.1, 0.2,...,0.9, as illustrated in Figure 4;
2. The swarm robots simulated have different memory lengths, ranging from 0, 50,...,900, giving 19 groups in total. In each group, there were 32 swarm robots each with different values for their hardware parameters.

The results for the simulated separation of swarm robots are shown in groups in Figure 5, and the following observations can be made.

When the swarm robots' memory length is zero, the swarm robots seldom separate in any of the arenas. During the simulations, the robots will try to follow the lines, with the *quasi-wind* maintaining the movement to the right of the arena. The swarm robots will quickly changes their orientation without any counteraction caused by the lack of any memory of previous experiences. This happens equally for all swarm robots in the group. In addition due to the large number of interactions as well as the randomness of the line orientations, all swarm robots have equal chance of orientating to any direction. Therefore in the end the swarm robots little or no separation is shown compared to other groups.

As the memory length increases, the swarm robots memorize their previous experiences which counteracts the *quasi-wind's* attempt to force the individual swarm robots to the right. As swarm robots in a group have different hardware variations, their memory weighting parameter will be different. The swarm robots act against the applied force differently: some swarm robots are able to maintain their previous orientation for sometime while some swarm robots are easily influenced and orientate themselves to move to the right. As expected the average speed of a swarm robots moving to the right of the arena is a function of their orientation over a period of time, causing the swarm robots to separate.

However when the memory length of the swarm robots further increase to 750, swarm robots in each group begin to converge. This is because the memory length of swarm robots is becoming excessive, the swarm robots always counteract to the *quasi-wind* even when they are off a line segment due to its large number of previous experiences memorized. Given a lengthy period of time, swarm robots in a group have equal chances of orientating to any direction, thus the speed of every swarm robot

have no significant differences. The result is that the separation of swarm robots in a group with large memory length is not as good as that of swarm robots with smaller values.

If we now consider the actual degree of separation, when the memory length of the swarm robots are small, not much difference can be seen in terms of the separation of the swarm robot within a group, as the average speeds for all robots in the X direction is almost identical. However as the memory length increases, individual swam robots begin to diverge within different arenas: swarm robots in the arenas with dense patterns begin so separate with in the same group, for instance when $density = 0.6$ and $ml = 450$, where some swarm robots in the group end their run at the far end of the arena and with some traveling not as far. This is because that when the density of the arena patterns are small, the number of interactions between swarm robots and the arena patterns are small, there are not enough patterns which can diversify the orientations of the swarm robots, leaving no chance for the long memory swarm robots to counteract with the *quasi-wind's* enforcing its orientation.

However when the arena density increases, more interactions occur and the previous experiences stored in the memory begin to oppose with the *quasi-wind's* attempt to maintain the swarm robots movement to the right. The longer memory length is, the more time it takes for the swarm robots to orientate straightly right (when $ml \leq 750$), during which swarm robot's right-forwards speed is reduced due to its orientation. However when the arena density further increase to $density > 0.6$, a tendency of convergence for the swarm robots' end locations can be seen, especially when $ml > 750$. This is because that as the density of the arena patterns is large and all swarm robots constantly encounter the line; the frequent interactions between swarm robots and arena patterns make swarm robots' orientations change constantly and leaving limited time for swarm robots' previous experiences to counteract against the *quasi-wind's* orientating effect. Therefore only a slight spread of the swarm robots' final location occurs.

6 CONCLUSIONS

In this work, we investigate the influence of swarm robot's memory length and the arena pattern density to the behavior sorting results in the context of an approach inspired by chromatography. Nineteen groups of hardware-varied swarm robots with different memory length are used in each of the nine arenas with different pattern densities, to investigate the amount of separation over a specific time. Results show that both memory length of swarm robots and arena pattern density are of significant importance to successful separation of the robots.

The length of the swarm robotic memory vector determines the length of the previous experiences that it can memorize. Without a sufficient amount of previous experiences, the swarm robots' orientation can be easily influenced by the *quasi-wind*
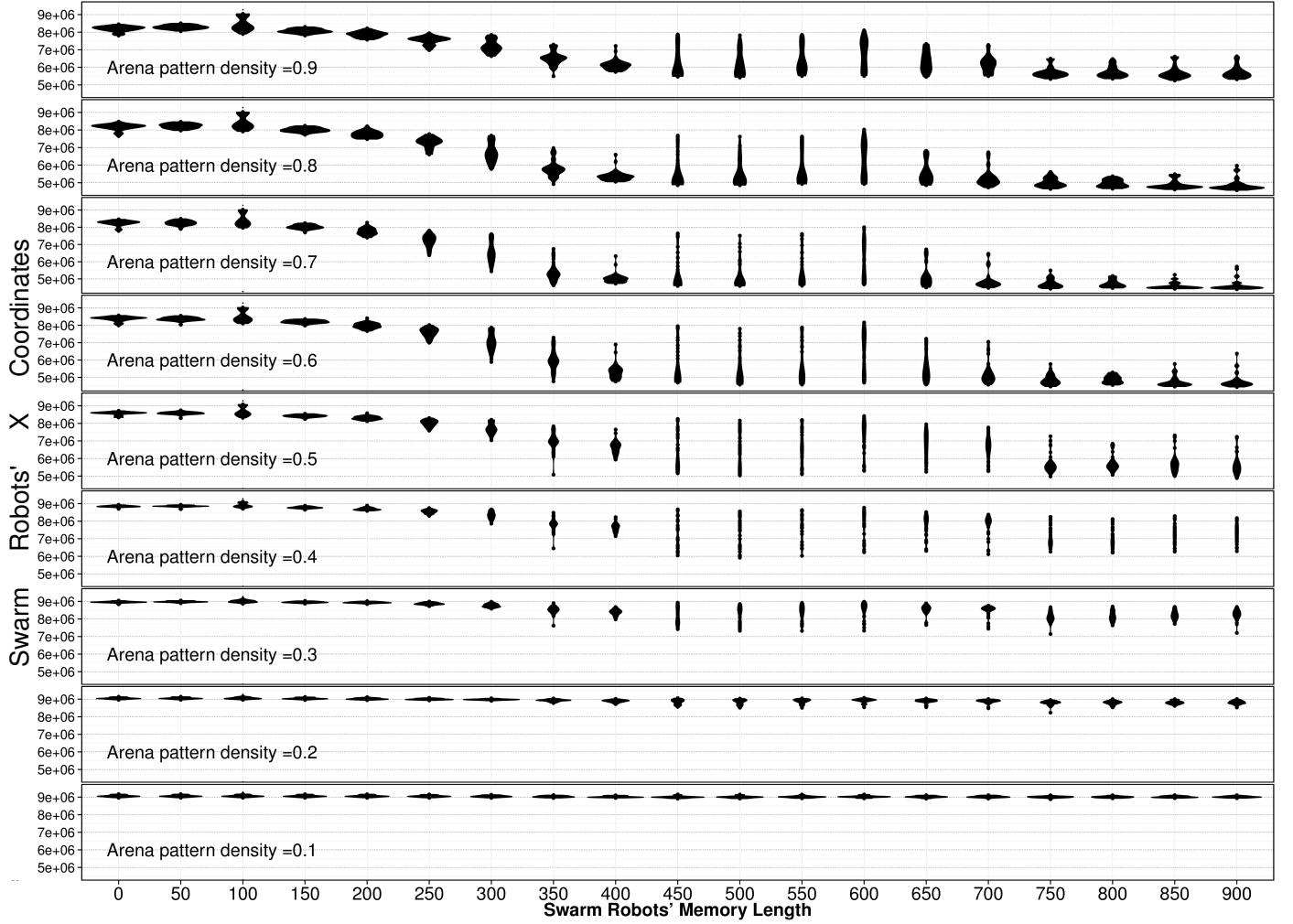
FIGURE 5. Violin plots illustrating the separation results of nineteen swarm robotic groups with different memory lengths. In each sub-figure, the results are shown for are nineteen groups of 32 robots, each group have unique length of memory. The memory length of the groups starts from 0 and increases by 50 steps per group until 900 steps. For each experiment, the width of the violin plot shows the number of robots converges at certain **X** coordinates. For example, in the experiment $ml = 0$ and $density = 0.1$, the violin plot is wide and its center is located at $y = 9e^6$, this means that the 32 robots that do not have any effective memory, locate in the arena with their X coordinates around $9e^6$, thus they do not separate well. However when $ml = 450$ and $density = 0.5$, the violin plot is narrow and long, it means that the 32 robots with 450 memory length separate equally on the X coordinates of the $density = 0.5$ chromatography arena with the range of ($5e^6$ to $8.5e^6$). The X co-ordinates are given in arbitrary units.

effect, leaving no opportunity for the swarm robots to show its behavioral characteristics. If the swarm robots remember too much, they all will be reluctant to interact with the arena pattern and orientating to the direction where they see fit, thus on average there is not much difference in their right-moving speed, resulting unsuccessful separation.

The density of the arena patterns determines the frequency of interactions between swarm robots and the environment. If the frequency is low, swarm robots' previous experiences have already been removed from the memory and the swarm robots keep moving to the right for for most of the time, cause no differ-

ence in their overall speed, resulting in unsuccessful separation. If the interaction frequency is too high, the distraction of the reflective pattern consistently changing the swarm robots' orientation without letting the robots show off their differences in terms of counteracting with the *quasi-wind* force, thus not much separation can be perceived.

Therefore memory length of the swarm robots and the arena pattern density has to match with each other in the swarm chromatography experiments for better sorting results. We view this findings as the further understanding of the swarm chromatography methods which improves the designing of behavioral sorting

experiments to achieve better separation of swarm robots. Future work will focus on further refining the separation of robotic behaviors, to allow individual specific behaviors to be identified.

REFERENCES

[1] Patil, M., Abukhalil, T., and Sobh, T., 2013. "Hardware Architecture Review of Swarm Robotics System: Self-Reconfigurability, Self-Reassembly, and Self-Replication". *ISRN Robotics,* **2013**, pp. 1–11.

[2] Borenstein, J., 1996. "Measurement and Correction of Systematic Odometry Errors in Mobile Robots". *IEEE Transactions on Robotics and Automation,* **12**, p. 6.

[3] Elliott, T., and Shadbolt, N. R., 2003. "Developmental robotics: Manifesto and application.". *Philosophical Transactions of the Royal Society of London, Series A,* **361**, pp. 2187–2206.

[4] Roth, Z., Mooring, B., and Bavani, B., 1987. "An Overview of Robot Calibration". *IEEE Journal of Robotics and Automation,* **3**(5), pp. 377–385.

[5] Lobo, J., and Dias, J., 2007. "Relative Pose Calibration Between Visual and Inertial Sensors". *International Journal of Robotics Research,* **26**(6), pp. 561–575.

[6] Crowder, R., and Zauner, K.-P., 2013. "A project-based biologically-inspired robotics module". *IEEE Transactions on Education,* **56**(1), pp. 82–87.

[7] Shang, B., Crowder, R., and Zauner, K.-P., 2013. "Simulation of Hardware Variations in Swarm Robots". In IEEE International Conference on Systems, Man, and Cybernetics, pp. 4066–4071.

[8] Shang, B., Crowder, R., and Zauner, K.-P., 2014. "Swarm Behavioral Sorting based on Robotic Hardware Variation". In The 5th International Conference on Simulation and Modeling Methodologies, Technologies and Application.

[9] Pugh, J., and Martinoli, A., 2007. "Parallel Learning in Heterogeneous Multi-Robot Swarms". In IEEE Congress on Evolutionary Computation, pp. 3839–3846.

[10] Alici, G., and Shirinzadeh, G., 2006. "A Systematic Technique to Estimate Positioning Errors for Robot Accuracy Improvement Using Laser Interferometry Based Sensing". *Mechanism and Machine Theory,* **40**, pp. 879–906.

[11] Korba, L., Elgazzar, S., and Welch, T., 1994. "Active Infrared Sensors for Mobile Robots". *IEEE Transactions on Instrumentation and Measurement,* **43**, pp. 283–287.

[12] Benet, G., Blanes, F., Simó, J., and Pérez, P., 2002. "Using Infrared Sensor for Distance MeasMeasure in Mobile Robots". *Robotics and Autonomous Systems,* **40**, pp. 255–266.

[13] Ettre, L., 1993. "Nomenclature for Chromatography". *Pure and Applied Chemistry,* **65**(4), pp. 819–872.

[14] Miller, J. M., 2009. *Chromatography: Concepts and Contrasts*. John Wiley & Sons: New Jersey.

[15] Sobol, I., and Levitan, Y., 1976. The production of points uniformly distributed in a multidimensional cube. Tech. Rep. 16, Institute of Applied Mathematics, USSR Academy of Sciences, January.