What a Sunflower Can Teach a Robot? Efficient Robot Queueing by Reverse Phyllotaxis

Yaroslav Litus and Richard T. Vaughan

Simon Fraser University, Burnaby, BC, Canada
ylitus@sfu.ca

Abstract

We present a distributed multi-robot controller for forming spatially efficient queues of arbitrary numbers of robots. The method is formally analyzed and validated in a conventional robot simulation. This controller is based on sunflower phyllotaxis and inherits its efficient packing properties. Two measures of queue spatial efficiency are proposed and their upper bounds for the presented controller are found. The controller compares favorably with a simple line queueing and shows unexpectedly high tolerance to spatial interference between robots.

Introduction

Living systems have evolved remarkable properties that are very desirable to have in embodied artificial systems including robots. Biomimetic robotics focuses almost exclusively on animals and bacteria, which is natural since members of these kingdoms face locomotion-related tasks similar to those of robots. In this paper we show that useful inspiration can be obtained from plants as their growth can be viewed as movement. We describe a multi-robot system based on phyllotaxis, in particular the arrangement of seeds on a sunflower head. To our knowledge, this is the first robot controller inspired by plant morphogenesis.

The problem we are solving is in the context of our interest in the energetics of large-scale multirobot teams. We believe that the ability to manage its own energy is a key characteristic of artificial and biological living systems. Autonomous energy management poses a plethora of challenges one of which is sharing a single charging station between many robots. Here we focus on finding an efficient way for robots to organize themselves into a queue while waiting for the service at the station.

Specifically, we are looking for a queue organization that will allow a large group of hungry robots to queue for the station without creating a major obstacle for other robots and without spending too much energy on supporting the formation. Thus, we want the queue to be dense and not to extend far in any direction so it is easy to navigate around. Also, we want to decrease the additional distance a robot needs to travel in order to join and move in the queue. Though we focus on robots and recharging, our arguments could be applied to any type of service and any embodied artificial living agents.

The next section reviews related work which is followed by definition of Vogel's sunflower phyllotaxis model. We present our modification of this model and define the robot controller based on the modified model. We analyze this controller in terms of two measures of queue spatial efficiency and compare it with a simple line queueing solution. After that we describe an informal experimental demonstration of the system and conclude by summarizing the paper and offering directions for future work.

Related work

Biomimetic robotics is a vibrant and diverse field. An up to date exploration of biomimetic robot mechanisms was done by Vepa (2009), while Bar-Cohen and Breazeal (2003) provide wider survey of the field and discus both mechanisms and control. Biologically inspired robot navigation was reviewed by Franz and Mallot (2000).

A rare plant-motivated robotics work by Armour and Vincent (2006) describes robot morphology motivated by tumbleweed plant. Another unconventional non-animal design is a robot controlled by a slime mold (Tsuda et al., 2007).

Phyllotaxis has been studied extensively both by mathematicians and biologists. One of the most known works on phyllotaxis modelling was done by Vogel (1979). An accessible introduction to the topic is available at Prusinkiewicz and Lindenmayer (1990, Ch. 4), while a detailed review of the early work was done by Jean (1994). Embryogenic mechanisms involved in phyllotaxis are disused in Traas and Vernoux (2002). In a recent work Nisoli et al. (2009) give experimental and numerical evidence for emergence of phyllotaxis in a system of repulsive particles.

Work on autonomous robot recharging has traditionally focused on the engineering issues of the problem and used simplistic non-optimal recharging policies (Silverman et al., 2002; Oh and Zelinsky, 2000). Recently Wawerla and Vaughan (2007) described a near-optimal robot recharging control policy that mimics animal foraging. Couture-Beil

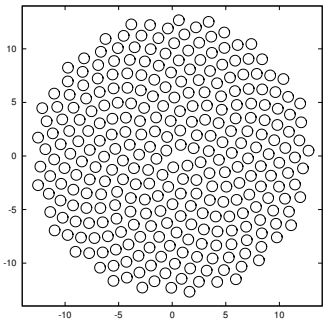


Figure 1: Sunflower head with 300 circular elements. Radius of an element is $r = 0.5$, Vogel constant $c = 0.75$.

and Vaughan (2009) developed an adaptive interference reduction strategy of placing recharging station and observed that the optimal location of the station is slightly off the path of working robots. A coordination mechanism for a large number of robots and multiple charging stations is presented by Drenner et al. (2009)

To our knowledge, no previous work has explicitly considered the cost of robot queues, either in terms of their direct navigation cost, or the indirect system cost due to the spatial interference they induce. Both are addressed here.

Efficient robot queueing by reverse phyllotaxis

Vogel's model

One of the best known models of sunflower phyllotaxis was proposed by Vogel (1979) in response to an early work by Mathai and Davis (1974). Vogel's model gives a constructive procedure for the shape of the mature sunflower head with the elements of equal sizes:

$$\rho = c\sqrt{n}, \tag{1}$$

$$\theta = ng, \tag{2}$$

where (ρ, θ) are the polar coordinates of the n-th element (r is the distance from the centre and θ is an angle between the element and a fixed axis passing through the centre), c is a scaling factor, and $g = \frac{2\pi}{\phi}$ is the golden angle (the smaller of the two angles produced by sectioning a circle circumference according to the golden ratio $\phi = \frac{1+\sqrt{5}}{2}$) so that the ratio of the full circumference to the larger arc is equal to the ratio of the larger arc to the smaller arc. A pattern produced by this model is shown on Fig. 1.

The elements in Vogel's model are arranged on a Fermat's (parabolic) spiral which has a general form $r^2 = a^2\theta$. Every turn of the spiral in the model contains on average $\phi + 1$ elements. Since Fermat's spiral crosses the annuli of equal areas in equal number of turns, equal areas on the head contain on average equal number of elements. The irrational angle between successive elements ensures that no two elements are located at the same angle. These two properties alone do not guarantee efficient packing of elements as locally the element packing density may differ significantly and large areas of unused space can be present.

However, the choice of the golden angle produces the theoretically most efficient packing of the elements among formations described by Eq. (1-2). Ridley (1982) proves that this angle will maximize the normalized packing efficiency defined as

$$\eta = A^{-1} \inf \left\{ |x - y|^2 : x, y \in X, x \neq y \right\},$$

where A is the average area occupied by each element (including its share of adjacent free space), and X is the set of elements. If η is high, then there are no areas where packing is too dense. Since elements are packed equally on average, having no overly dense areas ensures the absence of overly sparse areas with unused space.

This efficient packing and roughly circular shape of the sunflower head are appealing as a formation for a group of queueing robots. The head of the queue can be located at the centre of the sunflower and queuing robots can arrange themselves around it as if they were sunflower seeds. Simplicity of the model will transfer to the simplicity of a robot controller. Below we argue that this formation has a small diameter and allows for a low navigation overhead on joining and leaving the queue. However, first we need to provide a means for the robots to leave the queue once they were serviced.

Leaving the exit gap

Dense packing of the elements in Vogel's model makes the task of navigating from the centre of the formation outside very challenging. To minimize the interference and decrease the time spent on leaving the queue robots leave a gap from the centre of the formation to the periphery. This gap is located at a predefined angle and is wide enough for a robot to drive through (see Fig. 2). Assuming circular elements,

$$d(\rho, \theta) = |\rho \sin(\theta - \alpha)|, \tag{3}$$

$$V = \left\{ (\rho, \theta) | r = c\sqrt{n},\ \theta = gn \right\}, \tag{4}$$

$$G = \left\{ (\rho, \theta) \in V | (d(\rho, \theta) > s) \vee (\cos(\theta - \alpha) < 0) \right\}, \tag{5}$$

where $d(p)$ gives the distance of point p from the line passing through the centre of the exit gap, α is the direction angle of the gap, V is the set of element centres generated by Eq. (1-2), G is the set of element centres pruned of the elements that block the exit and s is the diameter of the element. The cosine condition in the generator for G is needed to restrict the blocking elements to the half plane in which the exit gap is located. As element locations are generated sequentially using Vogel's model, blocking elements can be skipped.

Therefore, leaving the queue amounts to simply going along the gap. Leaving the gap constantly open will increase

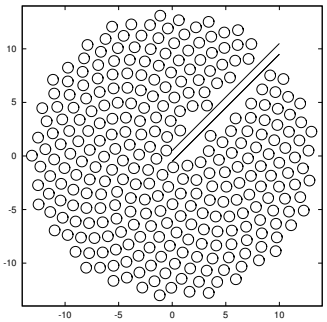


Figure 2: Sunflower head with 300 circular elements and an exit gap. Radius of an element is 0.5, Vogel constant $c = 0.75$. Exit direction $\alpha = \pi/4$.

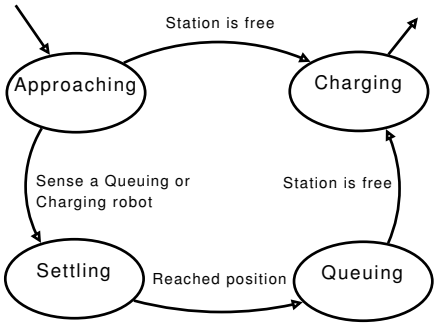


Figure 3: State diagram of the robot controller.

the diameter of the formation. However, it will eliminate the need for the queueing robots to move while letting a leaving robot through the formation. In the theoretical analysis section we explain why we believe this trade-off is reasonable. Also leaving a constant exit gap will be beneficial if the service rate is so high that a serviced robot starts leaving formation before another one finishes exiting.

Below we will use the terms "sunflower formation" and "sunflower queue" to refer to the formation with an exit gap unless specified otherwise.

Controller definition

We assume that robots are localized relative to the service stations. Every robot is equipped with a short range sensor capable of sensing the relative position of other robots. Possible choices for such a sensor include a stereo vision system and a laser-ranger-based fiducial finder. During the queueing routine a robot can be in one of four states. For simplicity we assume that if robots A senses robot B, it receives both relative position of B and its state. However, since the state of the robot can be deduced from its position and velocity, the state sensing is redundant. Sensors are subject to occlusions, so a robot can not sense through other robots.

Figure 3 describes the state diagram of the controller. When robot needs to get service, it switches into *Approaching* state. In this state robot drives straight to the charging station. If the station is free, it reaches it and switches to the *Charging* mode. Once recharged, the robot vacates the station and leaves along the predefined exit direction. If the station is busy, or the robot senses another robot in *Queueing* state, it switches into *Settling* state and calculates its position in queue based on the position of the furthest robot from the station observed so far. The Settling robot orbits around the queue and stops when it finds its position. Once there, the robot switches to Queueing state. If a robot in Queueing state is the closest one to the free charging station, it moves there and switches to Charging state. Other robots close to station sense this movement and move themselves closer to the station. This movement propagates through the whole queue and every robot moves closer to the station.

Below we provide a more formal description of behaviour in Settling, and Queueing states. All coordinates are polar with the origin located at the charging station. The currently assumed position of self on the formation is n, (ρ_d, θ_d) denotes the currently desired position, (ρ, θ) is the current actual position, ρ_i is the distance to station of the observed robot i (this can be calculated from the robot's own global position and the observed relative position of robot i), o is the orbiting offset, c is the Vogel constant, and g is the golden angle.

Settling A robot switches to the Settling state once it detects the presence of the queue by discovering that the charging station is occupied or sensing another robot in a Queueing state. The robot initializes its queue position to zero (line 1 of the Algorithm) and then processes positional information about the robot it senses.

The global position of a sensed robot is calculated as a sum of a global position of self and relative position of the sensed robot. If the robot observes a Queueing robot i the formation position of which is greater than the robots assumed position (line 4), the robot will chose the next position on a sunflower to occupy(line 5). If this position blocks the exit, the robot skips this position and chooses the next one (lines 6-8).

The robot calculates his desired coordinates from his selected position in the formation (lines 9-10) and navigates to that position in the following way. First, he moves from his current position to an orbit which is slightly above his desired distance from the charging station (line 13). Once on the orbit it moves on that orbit toward its desired angle (line 14). Once it successfully reaches this angle, it moves down from the orbit to the desired distance (lines 15-17).

The loop (2-18) ensures that the orbiting robot recomputes its desired position if it discovers a Queueing robot that occupies the desired position or even is further from it. At most one full turn around the formation will provide the Settling robot with a correct position in a formation. Once

the robot reaches its desired position, it switches to Queueing state(line 20).

Algorithm 1 Settling state controller

1: $n \leftarrow 0$
2: **repeat**
3: **for all** sensed robots i in $Queuing$ state **do**
4: **if** $\rho_i^2/c^2 \geq n$ **then**
5: $n \leftarrow \rho_i^2/c^2 + 1$
6: **if** $(c\sqrt{n}, ng)$ blocks the exit **then**
7: $n \leftarrow n + 1$
8: **end if**
9: $\rho_d \leftarrow c\sqrt{n}$
10: $\theta_d \leftarrow ng$
11: **end if**
12: **end for**
13: go to orbit $\rho_d + o$
14: move some distance along the circular orbit toward angle θ_d
15: **if** $\theta = \theta_d$ **then**
16: go to radius ρ_d
17: **end if**
18: **until** $p_c = p_d$
19: stop
20: $state \leftarrow Queuing$

Queueing A robot switches to Queueing state only from Settling state once the robot reaches its proper position in the formation. While in Queueing state, a robot finds the nearest robot it can sense that is closer to the charging station than itself and remembers the radius of this robot (line 1). If the robot senses a free charging station and is closer to the station than all queueing robots it senses to the station, then it is next to be charged and it proceeds to the station (lines 3-5). Once at the station the robot switches to Charging state.

The robot repeatedly finds the current value of the radius of the nearest sensed robot closer to the charging station (step 8). A change in the value means that a robot left the charging station, another robot occupied it and the queue moves closer to the station in response. This movement propagates from the centre of the formation to the periphery. The robot does not change its angle, but moves to the previous radius in the Vogel's model. Relocation happens once the new position is free (steps 9-11). If the new position blocks the exit the robot moves instead along the exit gap restoring the distance to the closest robot. When the robot reaches its new position, it updates the distance to the nearest robot which is closer to the charging station (step 16).

The positional update on steps 10-14 ensures that the order of recharging will correspond to the order of queueing. After the update the formation will remain a Vogel's formation with a gap. Such an update can be thought of as an inverse phyllotaxis during which the elements move inwards toward the centre instead of moving outwards.

Algorithm 2 Queueing state controller

1: $\rho_f \leftarrow \max_{\{\text{sensed Queuing } i|\rho_i<\rho\}} \rho_i$
2: **repeat**
3: $\rho_{\min} \leftarrow \min_{\{\text{sensed Queuing } i\}} \rho_i$
4: **if** $\rho < \rho_{\min}$ and charging station is free **then**
5: Move to charging station
6: $state \leftarrow Charging$
7: **else**
8: $\rho_c \leftarrow \max_{\{\text{sensed Queuing } i|\rho_i<\rho\}} \rho_i$
9: **if** $\rho_c < \rho_f$ and $(c\sqrt{(n-1)}, \theta)$ is free **then**
10: $n \leftarrow n - 1$
11: **if** $(c\sqrt{n}, \theta)$ is not blocking the exit **then**
12: Move to $(c\sqrt{n}, \theta)$
13: **else**
14: Move along the exit gap until $\rho_c = \rho_f$
15: **end if**
16: $\rho_f \leftarrow \max_{\{\text{sensed Queuing } i|\rho_i<\rho\}} \rho_i$
17: **end if**
18: **end if**
19: **until** $state = Charging$

Analysis

Our goal is to optimize two performance characteristics : the diameter of the queueing formation and the locomotion overhead on queueing. In this section we analyze the sunflower formation and provide theoretical guarantees of diameter and locomotion overhead. We will not make any assumptions about the initial spatial distribution of robots and service rate of charging station and derive instead the upper bounds of the performance characteristics. Moreover, since queueing overhead depends on the angle of the approach of the robot, worst case analysis allows to avoid complexities of parameterizing the result on that angle. Our primary interest is how performance characteristics change as the number of formation members grows.

Diameter Formation diameter is the maximum distance between the elements of the formation. Decreasing formation diameter is beneficial as it in general reduces the cost of non-participating robots to navigate around the formation.

Definition 1. *For a formation* $V = \{p|p \in \mathbb{R}^2\}$ *diameter* $d(V) = \max \|p_i - p_j\|, p_i \in V,\ p_j \in V$.

Lemma 1. *If* n *robots* s *are in the sunflower formation* $S(n)$ *with an exit gap,* $d(S(n)) \leq 2c\sqrt{2n} + s$*, where* c *is Vogel's constant and* s *is the size of the robot.*

Proof. Construction of formation with gap places elements according to the Eq.(1-2) skipping elements that block the exit. In Settling algorithm this skipping happens at steps

6-8. For a single-element-wide gap no two successive elements can block the exit, so at most every other position is skipped. Therefore, n-th element in the formation will be placed at most at radius $c\sqrt{2n}$. By construction, all previous elements are placed at smaller radii. Hence, the centres of all formation elements fit into a circle with diameter $2c\sqrt{2n}$. Since an element fits into the circle with diameter s, the maximum distance between points on formation surface is less than $2c\sqrt{2n} + s$. □

Locomotion overhead Queueing requires a robot to move into its position in the formation and then move in the queue until the robot reaches the charging station. This will usually require more locomotion than in the case where the station is free and robot can go straight for it. Locomotion overhead measures additional travelled distance caused by queueing.

Definition 2. *Locomotion overhead $P_o = P_r - P_s$, where $P_s = \|a - l\|$ is the distance between the point a at which robot detects the queue and starts a queueing manoeuvre, l is the location of charging stations, and P_r is the length of the robot trajectory from point a until it reaches the charging station while in queue* [1].

Lemma 2. *n-th robot in the queue has locomotion overhead $P_o(n) < (2\pi+2g)(c\sqrt{2n}+o)+c\sqrt{2n}-c\sqrt{2n-2}+o+2s$.*

Proof. Assume a robot detected a queue at point a. Its trajectory from that point to the charging station is comprised from three components (i) getting to the settling orbit, (ii) orbiting to the position in a queue, and (iii) moving toward the station while in a queue. By the argument used in the proof of Lemma 1 we conclude that n-th robot in a queue will settle at radius $c\sqrt{2n}$. The longest possible orbiting path for a robot n will result from detecting robot $n - 1$ only after one almost full turn around the queue and then skipping the next position on a spiral because it blocks the exit. Therefore, a robot will settle in less than one full turn and two golden angles on a circumference of circles of the radius less than $c\sqrt{2n} + o$, where o is the orbiting offset. Hence, component (ii) of the trajectory has an upper bound of $(2\pi + 2g)(c\sqrt{2n} + o)$.

Once settled and in a queue a robot moves only toward the charging station as it would do in the absence of a queue. Therefore, the only part of components (i) and (iii) that will contribute to the overhead is the travel from radius of point a to $c\sqrt{2n} + o$ and back. Because of the tight packing of the sunflower formation an approaching robot can travel at most one robot size s away from the outermost located robots before detecting the queue. That outermost located robot has number at least $2n - 2$. Therefore, the total contribution of (i) and (iii) is less than $s+o+c(\sqrt{2n}-\sqrt{2n-2})$ for a robot that is not encountering the exit gap on its straight path in a queue to the charging station.

For a case when robot has to follow the exit gap and depart from the straight path to the station a simple geometric argument shows that the increase in the path can not be greater than s. Hence, contribution of (i) and (iii) is bounded by $2s + o + c(\sqrt{2n} - \sqrt{2n-2})$ □

Comparison with the linear queueing There is no conventional robot queueing formation to serve as a benchmark for new queueing strategies. We will compare the sunflower formation with a simple and natural line queueing strategy. In this strategy a robot goes directly toward the charging station. If the charging station is occupied, the robot queues in a straight line that goes to the prespecified direction. To do this the robot follows the queue away from the station until it finds a free spot on a line. It is easy to argue, that diameter of this formation is $d(n) = ns$, where n is the number of robots in the queue and s is the size of the robot. Also, the locomotion overhead of n-th robot in line queueing is $P_o(n) = 2ns$ since the robot has to travel exactly two queue diameters before it reaches the charging station.

It seems that the linear queueing strategy has a lot of room for immediate improvement. For example, instead of going straight to the station, the robot can align itself with the queueing direction and then follow it to the station. If there is a queue, the robot detects it before reaching the station and can possibly reduce locomotion overhead by decreasing its travel to the tail of the queue. On the other hand, in case of no queue or a short queue this strategy will actually increase the overhead. A careful consideration shows that robot can make a correct decision on where to go only if he has an estimate of the current queue size beforehand. However, since the system described in this paper can also improve its performance by using a priori queue size information we will keep the comparison fair by using the simple uninformed linear queueing.

Linear queue diameters and the diameter bound of the sunflower queue differ in their rate of growth. The diameter of a linear queue grows linearly with queue cardinality, $d_l(n) = O(n)$, while the upper bound of a sunflower with a gap formation diameter grows at a slower square root rate $d_s(n) = O(\sqrt{n})$. Therefore, for any size of the robot and any Vogel's constant c the sunflower queue is guaranteed to eventually outperform the linear queue as the size of the queue grows, though for small queue this might not be the case.

Fig.4 compares the queue diameter of the simple linear queue with an upper bound of queue diameter of a sunflower formation for robots with size $s = 0.5$ and Vogel's constant

[1] It may be argued that the overhead should include leaving the formation and even returning to the original line of approach. However, it is not easy to define a standard way to measure these components of the trajectory across different formations. In any case, accounting for these components do not change the rate of growth of performance measure and qualitative comparison results we obtain.

$c = 0.6$. For queues with less than 13 robots a linear queue may have a smaller diameter, but for larger queues sunflower formation is guaranteed to outperform a linear queue. For 40 robots the sunflower queue already has half the linear queue diameter. The margin between the measures of two queues grows linearly as the queue size grows.

The locomotion overhead of the linear queue and the overhead bound of the sunflower queue relate similarly. The locomotion overhead of a linear queue grows linearly, while the overhead bound of a sunflower queue grows at a square root rate. Again, because of this difference in growth rates for any set of parameters there is a robot position for which the sunflower queue guarantees smaller overhead than the linear queue. For the larger robot positions sunflower queue will keep outperforming the linear queue and the margin between the measures will grow linearly with the robot position.

Fig.5 compares the locomotion overhead of the linear queue and the overhead upper bound of the sunflower queue for robots with size $s = 0.5$, Vogel's constant $c = 0.55$ and orbiting offset $o = 0.7$. For robot positions below 76 the linear queue can perform better, but for larger values the sunflower queue is guaranteed to have a smaller locomotion overhead.

Justification of leaving the exit gap Performance functions growth considerations can be also used to explain our choice of a leaving strategy for a recharged robot. Keeping the formation tight without a gap will lead to a constant factor improvement in the queue diameter, however robots will need to move and create an opening for every leaving recharged robot. Since the formation is tight, all robots will need to move whenever somebody leaves the formation from its centre. The last member of a queue with n members will need to move $O(n)$ times, therefore increasing the growth factor of the locomotion overhead from square root to linear. As we are interested in the efficient strategies for large number of robots, we prefer to leave a gap in a formation and suffer a constant factor increase in diameter but keep the growth rate of the locomotion overhead sublinear.

Demonstration

We implemented our queueing controller in the conventional robot simulator Stage (Vaughan, 2008). We simulate a team of 30 Pioneer robots in an 10m by 10m arena. Robots are equipped with short-range fiducial sensors capable of sensing bearing and distance to other robots, and a global positioning system. Robots do not communicate between themselves or with a charging station. Robots can collide, they have non-holonomic driving, and speed restriction and their fiducial sensors can be occluded by other robots. Parameters of simulation are given in Table 1.

We employ a simple orthodox reactive collision avoidance algorithm that uses range-finder readings. If there is an obstacle closer than a certain distance d_{stop}, the robot stops. If there is an obstacle which is at closer than a certain distance $d_{\text{avoid}} > d_{\text{stop}}$ then the direction that gave the smallest distance reading is found. If smallest reading came from the direction to the right of the robot bearing, a collision avoidance manoeuvre with a duration randomly selected in a certain interval is performed. The robot starts to turn left with a fixed turning speed and driving speed. Otherwise, the robot performs a right turn manoeuvre. If smallest reading came from the left, a right turn manoeuvre is performed. Once the collision avoidance manoeuvre is over, the robot continues to set the speed as prescribed by the main controller.

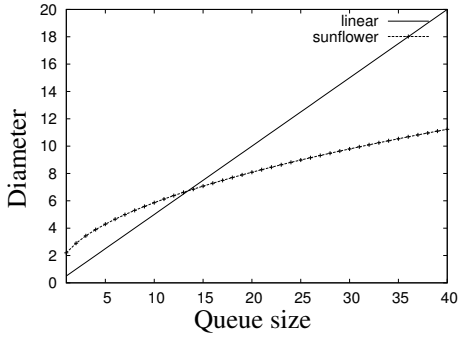


Figure 4: Diameters of the linear queue and the sunflower formation with a gap (vertical axis) plotted against number of robots in a queue (horizontal axis). Robot size $s = 0.5$, Vogel's constant $c = 0.6$.

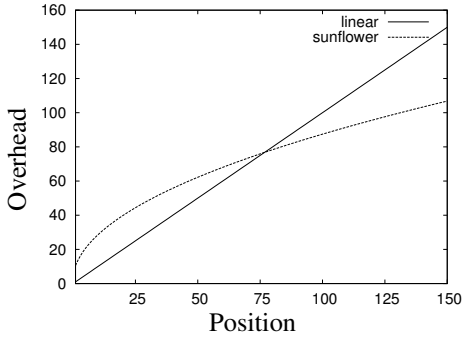


Figure 5: Locomotion overheads of the linear queue and the sunflower formation with a gap (vertical axis) plotted against robot position in queue (horizontal axis). Robot size $s = 0.5$, Vogel's constant $c = 0.55$, orbiting offset $o = 0.7$.

1: In the first set of simulations robots join the queue one by one with enough delay to let the previous robot settle in a queue and not create interference between settling robots.

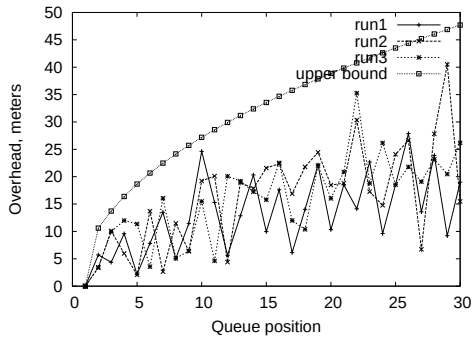


Figure 6: Locomotion overhead data from 3 experimental runs and a theoretical upper bound (vertical axis) plotted against robot position in queue (horizontal axis). Vogel's constant $c = 0.55$.

Locomotion overhead is measured as robots settle and move in queue. Once all robots join the queue, recharging start and robot recharge one by one until the queue is empty. The simulation stops once all robots are recharged. For various settings of Vogel's constant c and different initial approach of robots we observed the successful organization of sunflower formation with a gap and queue position updates after recharged robots depart.

Figure 6 shows the observed navigation overhead plotted against robot queue position from some representative example runs. All measured locomotion overheads were below the theoretically predicted upper bound (Lemma 2), which is also plotted. The angle of approach to the queue determined how closely the measured value approached the upper bound. If the approaching robot had the previously settled robot on the opposite side of the orbiting direction and beyond its sensor range, than an almost full turn around the formation was performed before the settling robot was able to sense it and calculate the position in formation. In this case the measured value of overhead was close to the theoretical upper bound. If the angle of approach allowed the robot to detect the last previously settled robot more quickly, then the measured value of overhead was significantly lower, than the predicted upper limits.

2: In the second set of simulations we tested how the system would cope with multiple robots approaching an empty queue at the same time. In this case they interfered with each other and a reactive collision avoidance algorithm took over control of the robots that came too close to other robots. The system handled interference unexpectedly well. For a small number of simultaneously approaching robots (between two and five) the system reliably created the formation albeit with a delay caused by repeated interference avoidance. For larger number of simultaneously approaching robots occasional collisions were observed as the collision avoidance was not able to handle large number of robots in close proximity to each other. However, most of the collisions were resolved by the emergent "helping" behaviour of other robots that approached stuck robots and triggered collision avoidance that separated them. Even for a very large number of robots successful formation creation was possible.

Maximum speed	0.4 m/s
Collision avoidance speed	0.05 m/s
Collision avoidance turning speed	0.5 rad/s
Collision avoidance initiation distance	0.6 m
Minimum front stopping distance	0.5 m
Collision avoidance duration interval	[1,2]s
Fiducial finder range	2 m
Orbiting offset	0.7 m
Position settling precision	0.05 m

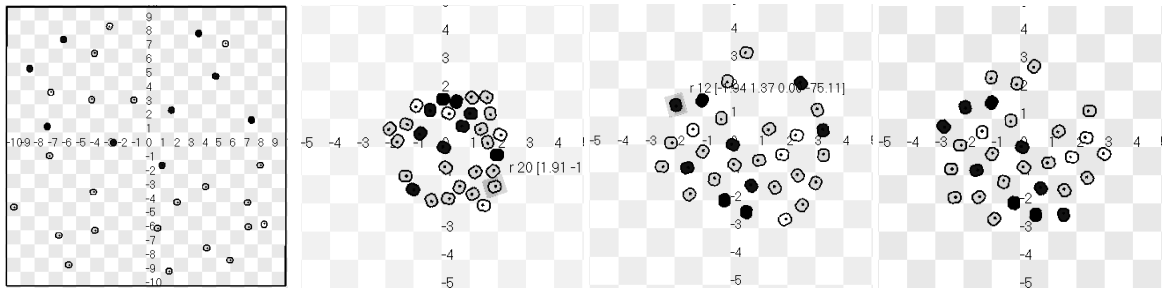
Table 1: Parameters used in Stage simulation

Fig. 7 illustrates successful creation of the formation by the group of 30 robots. Fig. 7(a) shows the initial positions of the robots. As they all simultaneously drive for the charging station a lot of interference occurs and robots spend most of the time in collision avoidance mode (See Fig. 7(b) with two robots in position and the rest interfering with each other). Eventually robots succeed in settling in positions and formation starts to grow (see Fig. 7(c) with 8 robots still settling). Fig. 7(d) shows the final formation.

Observe the group of robots following each other on the orbit in the right side of Fig. 7(c). This emergent "train formation" behaviour results from the interaction of orbiting part of the settling algorithm and the collision avoidance mechanism that randomizes the collision avoidance manoeuvre duration thus spreading robots in time. We believe that this emergent behaviour explains the tolerance of the system to spatial interference. As the queue forms, the system is capable of handling increasing numbers of simultaneously joining robots as the orbit circumference increases.

Conclusion

The focus of this paper was on autonomous creation of spatially efficient queues by a group of robots. We described a novel distributed decentralized queue formation algorithm inspired by the plant phyllotaxis, which we call the *sunflower formation*. To our knowledge this is the first robot control algorithm inspired by phyllotaxis. We defined two measures of spatial efficiency for robot queues and proved upper bounds of these measures for the sunflower formation algorithm. Our algorithm compares favourably with a simple linear queueing algorithm showing superior asymptotic behaviour of both measures. The controller was successfully



(a) (b) (c) (d)

Figure 7: 30 robots simultaneously attempting to join an empty queue. Vogel's constant $c = 0.55$ Exit direction $\alpha = \pi/3$.

demonstrated in a conventional multi-robot simulation and showed an unexpectedly high spatial interference tolerance.

This work can be extended in many directions. The first is the extension of the algorithm to create efficient queue formation in three dimensions with potential application in aerial, space and underwater robotics. A second direction is looking for ways to improve queueing as a system component, for example integrating it with a custom collision avoidance algorithm that would favour its emergent properties and allow it to successfully manage larger number of simultaneously approaching robots. Also, it may be possible to eliminate the need for global localization by using the relative poses of sensed queueing robots in addition to their relative positions. Finally, other queueing formation like zig-zag queue and theoretically optimal hexagonal packing should be investigated.

A very interesting direction is looking for ways to base the controller on models of emergent phyllotaxis instead of the constructive model employed here. Finally, we believe that plant kingdom has a lot of hidden potential for biomimetic robotics that is waiting to be discovered and exploited.

References

Armour, R. H. and Vincent, J. F. (2006). Rolling in nature and robotics: A review. *Journal of Bionic Engineering*, 3(4):195 – 208.

Bar-Cohen, Y. and Breazeal, C., editors (2003). *Biologically-Inspired Intelligent Robots*. SPIE- International Society for Optical Engineering.

Couture-Beil, A. and Vaughan, R. T. (2009). Adaptive mobile charging stations for multi-robot systems. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS'09)*, St. Loius, MO.

Drenner, A., Janssen, M., and Papanikolopoulos, N. (2009). Coordinating recharging of large scale robotic teams. In *IROS'09: Proceedings of the 2009 IEEE/RSJ international conference on Intelligent robots and systems*, pages 1357–1362, Piscataway, NJ, USA. IEEE Press.

Franz, M. and Mallot, H. A. (2000). Biomimetic robot navigation. *Robotics and autonomous Systems*, 30:133–153.

Jean, R. V. (1994). *Phyllotaxis: A Systemic Study in Plant Morphogenesis*. Cambridge University Press.

Mathai, A. and Davis, T. (1974). Constructing the sunflower head. *Mathematical Biosciences*, 20(1-2):117 – 133.

Nisoli, C., Gabor, N. M., Lammert, P. E., Maynard, J. D., and Crespi, V. H. (2009). Static and dynamical phyllotaxis in a magnetic cactus. *Phys. Rev. Lett.*, 102(18):186103.

Oh, S. and Zelinsky, A. (2000). Autonomous battery recharging for indoor mobile robots. In *Proceedings of the Australian Conference on Robotics and Automation*.

Prusinkiewicz, P. and Lindenmayer, A. (1990). *The algorithmic beauty of plants*. Springer-Verlag New York, Inc., New York, NY, USA.

Ridley, J. (1982). Packing efficiency in sunflower heads. *Mathematical Biosciences*, 58(1):129 – 139.

Silverman, M., Nies, D. M., Jung, B., and Sukhatme, G. S. (2002). Staying alive: A docking station for autonomous robot recharging. In *IEEE International Conference on Robotics and Automation*, pages 1050–1055, Washington D.C.

Traas, J. and Vernoux, T. (2002). The shoot apical meristem: the dynamics of a stable structure. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 357(1422):737–747.

Tsuda, S., Zauner, K.-P., and Gunji, Y.-P. (2007). Robot control with biological cells. *Biosystems*, 87(2-3):215 – 223. Papers presented at the Sixth International Workshop on Information Processing in Cells and Tissues, York, UK, 2005 - IPCAT 2005, Information Processing in Cells and Tissues.

Vaughan, R. T. (2008). Massively multi-robot simulations in Stage. *Swarm Intelligence*, 2(2-4):189–208.

Vepa, R. (2009). *Biomimetic Robotics: Mechanisms and Control*. Cambridge University Press.

Vogel, H. (1979). A better way to construct the sunflower head. *Mathematical Biosciences*, 44(3-4):179 – 189.

Wawerla, J. and Vaughan, R. T. (2007). Near-optimal mobile robot recharging with the rate-maximizing forager. In *Proceedings of the European Conference on Artificial Life (ECAL)*, pages 776–785, Lisbon, Portugal.