Swarm Intelligence manuscript No.
(will be inserted by the editor)

Implementing Continuous Flow of Information Using Centralized Communication for a Robotic Validation of the Attractive Field Model

Md Omar Faruque Sarker · **Torbjørn S. Dahl**

the date of receipt and acceptance should be inserted later

Abstract Attractive field model is an interdisciplinary model of self-organization that proposes to solve the issue of division of labour or task allocation using a set of generic rules derived from the observation of ant, human and robotic social systems. These bottom-up rules can be used to model multi-robot task allocation problem in terms of attractive fields between robots and tasks. The concrete form of these rules provides sufficient abstraction to accommodate different sensing and communication model. By avoiding strong dependence on local interactions found in many existing approaches to multi-robot task allocation, attractive field model becomes flexible enough to achieve task allocation among multiple concurrent tasks through the presence of a system-wide continuous flow of information. Unlike traditional swarm task-allocation which extensively relies upon local communication, our approach uses a centralized communication scheme to obtain continuous flow of information among variable number of independent task-allocating robots. Our experimental results validate the effectiveness of using attractive filed model with a centralized communication scheme as a mechanism for self-organized multi-robot task allocation. Our experiments used 8 and 16 e-puck robots in 1m x 1m and 2m x 2m areas respectively.

1 Introduction

Multi-robot task allocation (MRTA) problem focuses on allocating appropriate tasks to appropriate robots dynamically considering the changes in task-requirements, team-performance and environment. This problem is analogous to the division of labour among individuals found in biological and human social systems. MRTA is a challenging research issue since there exist no central planner or coordinator in a distributed multi-robot team and robots have limited capabilities to sense, to communicate and to interact with a limited number of peers at a time. Due to large communication and computational overhead, traditional explicit task-allocation approaches, such as intentional cooperation (Parker, 2008) and market-based bidding approach (Dias et al, 2006), do not scale well to the large number of tasks and robots (Lerman et al, 2006). On the other hand, existing self-organized approaches, such as emergent cooperation Kube and Zhang (1993), use of *adaptation rules* (Liu et al, 2007)

Cognitive Robotics Research Centre, University of Wales, Newport, UK
E-mail: Mdomarfaruque.Sarker | Torbjorn.Dahl@newport.ac.uk

are limited since they produce specific solutions to certain global tasks alone (Gerkey and Mataric, 2004).

As the part of the Engineering and Physical Sciences Research Council (UK) collaborative research project, 'Defying the Rules: How Self-regulatory Systems Work', we have studied the behaviour of ants, humans and robots and formalized a set of necessary and sufficient requirements for self-organisation in social systems. These four requirements are: *continuous flow of information*, *concurrency*, *learning* and *forgetting* (which are explained later). Primarily, these requirements facilitate the derivation of local control rules for regulating an individual's task-allocation behaviour in such a way that can develop self-organized division of labour in the entire group. However unlike most of the existing self-organized task-allocation approaches, these requirements do not assume the presence of any specific type of interaction or communication pattern of the group. Rather the continuous flow of information propagates the task-requirements to all robots. This flow of information can provide robots with necessary perception about task-requirements as well as the feedback about their collective performance. This flow of information can happen by global broadcast, local peer-to-peer interaction or even through stigmergy or a shared global memory. Our model does not assume any fixed communication set-up. Thus it can accommodate various forms of communication and interactions among individuals and the environment that collectively contribute to generate a continuous flow of information in the system.

This article presents our works on the validation of the effectiveness of AFM in producing self-organized MRTA using a centralized communication scheme. The main contributions of this paper are: interpretation of AFM for a multi-robot system, implementation of the first requirement or the continuous flow of information and last, but not the least, validation of AFM in achieving self-organized MRTA in a fairly large multi-robot system. Rest of this article is organized as follows. Section 2 presents the interpretation of AFM for multi-robot systems. Our centralized communication scheme is presented in Section 3. Section 4 presents the design of our experiments including specific parameters and observables. Section 5 discusses our experimental results from 8 and 16 robot experiments. Section 7 reviews related work and the correspondence of AFM with biological self-organization. Section 8 draws conclusions..

2 Robotic interpretation of the Attractive Field Model

The AFM has been developed through the collaborative effort among the partners of our EPSRC project which include experiments on *Temnothorax albipennis* ant colonies to study the collective performance of ants during brood-sorting and nest construction after emigration to a new site. The model is also inspired by observational data from the self-organized development of the infrastructure of an *eco-village* by an open community of volunteers. These studies helped us to formalize the a set of necessary and sufficient requirements for self-organisation in a social systems as discussed before. Below we discuss the generic framework of self-regulated division of labour under AFM and our interpretation of this framework for multi-robot systems.

2.1 The Attractive Field Model

The first requirement of self-organization, i.e. continuous flow of information, demands that self-organised social systems must establish a flow of information over the period of time

when self-organisation can be defined. The task information provides the basis on which the individuals self-organise by enabling them to perceive tasks and receive feedback on collective team performance. Concurrence or the simultaneous presence of several task-options is the second requirement of self-organization. This is necessary in order to meaningfully say that the system has organised into a recognisable structure. In task-allocation terms, the minimum requirement is a single task as well as the option of not performing any task. The third requirement of self-organization is the presence of different levels of sensitization among individuals towards different tasks. A system, where each robot has different levels of *sensitization* to the available tasks, can embody a distinct organisation through differentiation. Finally, the last requirement of self-organization is to implement forgetting of tasks among individuals. In order to avoid saturation, forgetting mechanism reduces the sensitisation levels of individuals towards different tasks. This allows flexibility in the system as the individual preferences towards different tasks always keep changing over time.

Based on the above requirements of self-organization we have developed a formal model of self-organized division of labour in social systems named the *attractive field model* (AFM) (Arcaute et al, 2008). We have selected the issue of division of labour since this is a well-understood problem in biological and human social systems. Now let us at first see how division of labour in ants, humans and robots can be explained in terms of the above four requirements. In an ant colony, we have assumed that division of labour is not genetically driven. Initially all ants are equal. It is not well-known how ants "know" or get information about tasks. But we know that they all interact directly and indirectly and perform tasks. The flow of information can take place in many different ways. Ants can get information through direct peer-to-peer, local or global broadcast and indirect pheromone communications. Thus each task can be treated as an attractive source, stimulating ants to go to it. Here the stimulus primarily depends on the distance. In case of sensitization or learning, an ant that has performed a task, it is assumed that it will be more likely that it will be attracted to do it again. Concurrence of tasks can also be achieved through spatial dependence of strength of stimulus. Some ants will be favoured to get information from multiple sources than others. When an ant does not do a task for relatively long time it is less probable that it will do that task again. This will lead to forgetting of their tasks gradually.

Similarly, we can see that in human society people can get information about certain tasks or resources from various sources. Those who are located near the source of information are more likely to attend it. For example, trough the access to Internet, some people can get information quicker than others. The background training, education or skill profile can be used to estimate the sensitization to do a task or use a resource.

Building on the requirements for self-organised social systems, AFM formalises these requirements in terms of the relationships between properties of individual agents and of the system as a whole (Arcaute et al, 2008). AFM is a bipartite network, i.e. there are two different types of nodes. One set of nodes describes the sources of the attractive fields, the tasks, and the other set describes the agents. Edges only exist between different types of nodes and they encode the strength of the attractive field as perceived by the agent. There are no edges between agent nodes. All communication is considered part of the attractive fields. There is also a permanent field representing the *no-task* option of not working in any of the available tasks. This option is modelled as a random walk. AFM is presented graphically in Fig. 1. The elements depicted are:

1. Source nodes (o) are tasks to be allocated to agents
2. Agent nodes (x) e.g., ants, humans, or robots

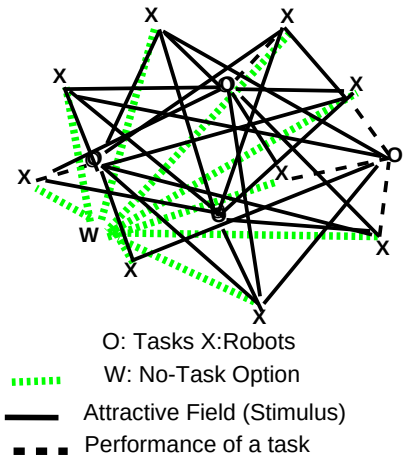

Fig. 1 Attractive Filed Model (AFM)

3. Black solid edges represent the attractive fields and correspond to an agent's perceived stimuli from each task.
4. Green edges represent the attractive field of the ever present no-task option, represented as a particular task (w).
5. The red lines are not edges, but represent how each agent is allocated to a single task at any point in time.

The edges of the AFM network are weighted and the value of this weight describes the strength of the stimulus as perceived by the agent. In a spatial representation of the model, the strength of the field depends on the physical distance of the agent to the source. In information-based models, the distance can represent an agent's level of understanding of that task. The strength of a field is increased through the sensitisation of the agent through experience with performing the task. This elements is not depicted explicitly in Figure 1 but is represented in the weights of the edges. In Figure 1), the nodes have arbitrary positions. Even though the distance is physical in this case, it need not be. When the model is applied to other domains, the distance can represent the accessibility of information or the time the information takes to reach the agent. In summary, from the above diagram of the network, we can see that each of the agents is connected to each of the tasks. This means that even if an agent is currently involved in a task, the probability that it stops doing it in order to pursue a different task, or to random walk, is always non-zero.

AFM assumed a repeated task selection by individual agents. The probability of an agent choosing to perform a task is proportional to the strength of the task's attractive field, as given by Equation 1.

$$P_j^i = \frac{S_j^i}{\sum_{j=0}^{J} S_j^i} \quad where, \quad S_0^i = S_{RW}^i \tag{1}$$

Equation 1 states that the probability of an agent, i, selecting a task, j, is proportional to the stimulus, S_j^i, perceived from that task, with the sum of all the task stimuli normalised to 1.

The strength of an attractive field varies according to how sensitive the agent is to that task, k_j^i, the distance between the task and the agent, d_{ij}, and the *urgency*, ϕ_j of the task. In order to give a clear edge to each field, its value is modulated by the hyperbolic tangent

function, *tanh*. Equation 2 formalises this part of AFM.

$$S_j^i = tanh\{\frac{k_j^i}{d_{ij}+\delta}\phi_j\} \tag{2}$$

Eqation 2, used small constant δ, called *delta distance*, to avoid division by zero, in the case when a robot has reached to a task.

Equation 3 shows how AFM handles the the no-task, or random walk, option. The strength of the stimuli of the random walk task depends on the strengths of the fields real tasks. In particular, when the other tasks have a low overall level of sensitisation, i.e., relatively weak fields, the strength of the random walk field if relatively high. On the other hand, when the agent is highly sensitised, the strength of the random walk field becomes relatively low. We use J to denote the number of real tasks. AFM effectively considers random walking as an ever present additional task. Thus the total number of tasks becomes $J+1$.

$$S_{RW}^i = tanh\left\{1 - \frac{\sum_{j=1}^{J} S_j^i}{J+1}\right\} \tag{3}$$

A task j has an associated urgency ϕ_j indicating its relative importance over time. If an agent attends a task j in time step t, the value of ϕ_j will decrease by an amount $\delta_{\phi_{INC}}$ in the time-step $t+1$. On the other hand, if a task has not been served by any of the agents in time-step t, ϕ_j will increase by a different amount, $\delta_{\phi_{DEC}}$ in time-step $t+1$. This behaviour is formalised in Equations 4 and 5.

$$If\ the\ task\ is\ not\ being\ done:\ \ \phi_{j,t+1} \rightarrow \phi_{j,t}\ \ + \delta_{\phi_{INC}} \tag{4}$$

$$If\ the\ task\ is\ being\ done:\ \ \phi_{j,t+1} \rightarrow \phi_{j,t}\ \ - n\ \delta_{\phi_{DEC}} \tag{5}$$

Equation 4 refers to a case where no agent attends to task j and Equation 5 to the case where n agens are concurrently performing task j.

In order to complete a task, an agent needs to be within a fixed distance of that task. When an agent performs a task, it learns about it and this will increases the probability of that agent selecting that task in the future. This is done by increasing its sensitization to the task by a fixed amount, k_{INC}. The variable affinity of an agent, i, to a task, j, is called its *sensitization* to that task and is denoted k_j^i. If an agent, i, does not do a task j, k_j^i is decreased by a different fixed amount, k_{DEC}. This behaviour is formalised in Equations 6 and 7.

$$If\ task\ is\ done:\ \ k_j^i \rightarrow k_j^i\ \ +\ \ k_{INC} \tag{6}$$

$$If\ task\ is\ not\ done:\ \ k_j^i \rightarrow k_j^i\ \ -\ \ k_{DEC} \tag{7}$$

The interpretation of AFM in a multi-robot system follows the above mentioned generic interpretation. Each robot can be modelled as an agent and each task can be modelled as a spatial location. Depending on a specific task performance, sensitization levels of robots for that task can be increased or decreased. Robots can have multiple options to select a task from a set of available tasks. If there is only a global task, e.g. in case of foraging task, random walking can be used as a no-task option. Thus using this abstract approach, we can effectively model the MRTA issue of a global task as well as multiple concurrent tasks in a multi-tasking environment. The distance between a task and a robot is simply the physical distance and the sensitivities are recorded as specific values on each robot. The urgency values of the tasks are calculated based on the number of robots attending each task and the updated urgency values are communicated to the robots.

The sensing of the distance between the taks and robots as well as the communication of urgency values are non-trivial in a robotic systen. Both the sensing and communication can be done either locally by the individual robots or centrally, through an overhead camera and a global communication network. This article presents work on exploring the effects of different sensing and communication models on the performance of MRTA systems.

2.2 Task-allocation in a Manufacturing Shop-Floor

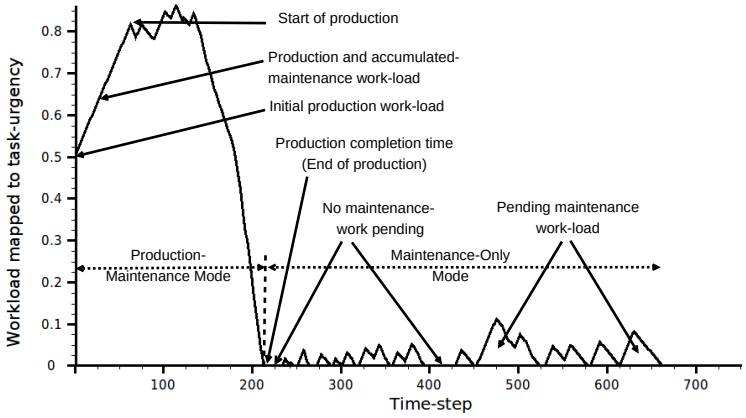

Fig. 2 Virtual Shop-floor production and maintenance cycle

We have designed a set of manufacturing shop-floor scenario for validating the effectiveness of AFM in producing self-regulated MRTA. By extending our interpretation of AFM in multi-robot system, we can set-up manufacturing shop-floor scenario. Here,, each task represents a manufacturing machine. These machines are capable of producing goods from raw materials, but they also require constant maintenance works for stable operations. Let W_j be a finite number of material parts that can be loaded into a machine j in the beginning of its production process and in each time-step, ω_j units of material parts can be processed ($\omega_j \ll W_j$). So let Ω_j^p be the initial production workload of j which is simply: W_j/ω_j unit. We assume that all machines are identical. In each time step, each machine always requires a minimum threshold number of robots, called hereafter as *minimum robots per machine (μ)*, to meet its constant maintenance work-load, Ω_j^m unit. However, if μ or more robots are present in a machine for production purpose, we assume that, no extra robot is required to do its maintenance work separately. These robots, along with their production jobs, can do necessary maintenance works concurrently. For the sake of simplicity, in this paper we consider $\mu = 1$. Now let us fit the above production and maintenance work-loads and task performance of robots into a unit task-urgency scale. Let us divide our manufacturing operation into two subsequent stages: 1) *production and maintenance mode (PMM)*, and 2) *maintenance only mode (MOM)*. Initially a machine starts working in PMM and does production and maintenance works concurrently. When there is no production work left, it then enters into MOM. Fig. 2 illustrates this for a single machine. Under both modes, let α_j be the amount of workload occurs in a unit time-step if no robot serves a task and it corresponds to a fixed task-urgency $\Delta\phi_{INC}$. On the other hand, let us assume that in each time-step, a

robot, i, can decrease a constant workload β_i by doing some maintenance work along with doing any available production work. This corresponds to a negative task urgency: $-\Delta\phi_{DEC}$. So, at the beginning of production process, task-urgency, occurred in a machine due to its production work-loads, can be encoded by Eq. 8.

$$\Phi_{j,INIT}^{PMM} = \Omega_j^p \times \Delta\phi_{INC} + \phi_j^{m0} \tag{8}$$

where ϕ_j^{m0} represents the task-urgency due to any initial maintenance work-load of j. Now if no robot attends to serve a machine, each time-step a constant maintenance workload of α_j^m will be added to j and that will increase its task-urgency by $\Delta\phi_{INC}$. So, if k time steps passes without any production work being done, task urgency at k^{th} time-step will follow Eq. 9.

$$\Phi_{j,k}^{PMM} = \Phi_{j,INIT}^{PMM} + k \times \Delta\phi_{INC} \tag{9}$$

However, if a robot attends to a machine and does some production works from it, there would be no extra maintenance work as we assumed that μ = 1. Rather, the task-urgency on this machine will decrease by $\Delta\phi_{DEC}$ amount. If ν_k robots work on a machine simultaneously at time-step k, this decrease will be: $\nu_k \times \Delta\phi_{DEC}$. So in such cases, task-urgency in $(k+1)^{th}$ time-step can be represented by:

$$\Phi_{j,k+1}^{PMM} = \Phi_{j,k}^{PMM} - \nu_k \times \Delta\phi_{DEC} \tag{10}$$

At a particular machine j, once $\Phi_{j,k}^{PMM}$ reaches to zero, we can say that there is no more production work left and this time-step k can give us the *production completion time* of j, T_j^{PMM}. Average production time-steps of a shop-floor with M machines can be calculated by the following simple equation.

$$T_{avg}^{PMM} = \frac{1}{M}\sum_{j=0}^{M} T_j^{PMM} \tag{11}$$

T_{avg}^{PMM} can be compared with the minimum number of time-steps necessary to finish production works, T_{min}^{PMM}. This can only happen in an ideal case where all robots work for production without any random walking or failure. We can get T_{min}^{PMM} from the total amount of work load and maximum possible inputs from all robots. If there are M machines and N robots, each machine has Φ_{INIT}^{PMM} task-urgency, and each time-step robots can decrease N $\times$ $\Delta\phi_{DEC}$ task-urgencies, then the theoretical T_{min}^{PMM} can be found from the following Eq. 12.

$$T_{min}^{PMM} = \frac{M \times \Phi_{INIT}^{PMM}}{N \times \Delta\phi_{DEC}} \tag{12}$$

$$\zeta_{avg}^{PMM} = \frac{T_{avg}^{PMM} - T_{min}^{PMM}}{T_{min}^{PMM}} \tag{13}$$

Thus we can define ζ_{avg}^{PMM}, *average production completion delay* (APCD) by following Eq. 13: When a machine enters into MOM, only μ robots are required to do its maintenance works in each time step. So, in such cases, if no robot serves a machine, the growth of task-urgency will follow Eq. 9. However, if ν_k robots are serving this machine at a particular time-step k^{th} , task-urgency at $(k+1)^{th}$ time-step can be represented by:

$$\Phi_{j,k+1}^{MOM} = \Phi_{j,k}^{MOM} - (\nu_k - \mu) \times \Delta\phi_{DEC} \tag{14}$$

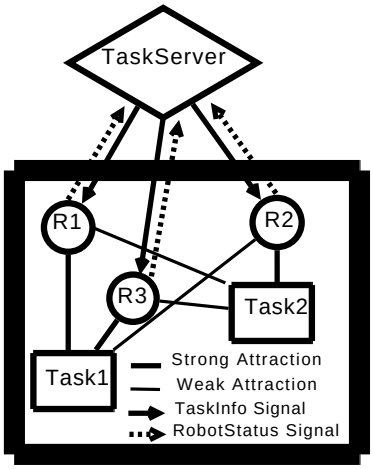

Fig. 3 A centralized communication scheme

By considering $\mu = 1$ Eq. 14 will reduces to Eq. 10. Here, $\Phi_{j,k+1}^{MOM}$ will correspond to the *pending maintenance work-load (PMW)* of a particular machine at a given time. This happens due to the random task switching of robots with a no-task option (random-walking). Interestingly PMW will indicate the robustness of this system since higher PMW value will indicate the delay in attending maintenance works by robots. We can find the average PMW (APMW) per time-step per machine, χ_j^{MOM} (Eq. 15) and average PMW per machine per time-step, χ_{avg}^{MOM} (Eq. 16).

$$\chi_j^{MOM} = \frac{1}{K} \sum_{k=1}^{K} \Phi_{j,k}^{MOM} \qquad (15)$$

$$\chi_{avg}^{MOM} = \frac{1}{M} \sum_{j=1}^{M} \chi_j^{MOM} \qquad (16)$$

3 Centralized Communication Scheme

AFM relies upon a system-wide continuous flow of information which can be realized using any suitable communication model. A simple centralized communication scheme is outlined in Fig. 3. In this model we have used bi-directional signal-based exchange of communication messages between a centralized *task perception server* (TPS) and a *robot-controller client* (RCC). The main role of TPS is to send up-to-date task-information to RCCs. This task-information mainly contains the location and urgency of all tasks which is used by the RCCs for running their task-allocation algorithm. The urgency value of each task is dynamically updated by TPS after receiving the status signals from the working robots of that particular task. Fig. 3 shows how three robots are attracted to two different tasks and their communications with TPS.

We can characterize our communication model in terms of three fundamental issues of communication: i) message content, ii) communication frequency and iii) target message recipients (Gerkey and Mataric, 2001). AFM suggests the communication of task-urgencies among robots. This communication helps the robots to gain information that can be treated as “global sensing”. However in this model robots do not communicate among themselves.

Since in order to run the task-allocation algorithm robot-controllers need the distance information we also include the task position information in the message. Our centralized communication model is open to include any further information, such as time-stamp, in the message payload. In this model the frequency of signal emission depends on several issues, e.g. the rate at which the environment is changing, the bandwidth of communication medium. In case of time-extended tasks, robots can receive information less frequently and the bandwidth usage can be kept minimum. However under a fast changing environment relatively more bandwidth will be required. Finally the centralized communication model spread the attractive fields of all tasks globally by broadcasting information to all robots.

4 Experiments

In this section, we have described the design of parameters and observables of our experiments within the context of our manufacturing shop-floor scenario. These experiments are designed to validate AFM by testing the presence of division of labour, such task specialization, dynamic task-switching or plasticity etc.

4.1 Observables

Plasticity: Self-regulated MRTA is often characterised by the plasticity and task-specialization, in both macroscopic and microscopic levels. Within our manufacturing shop-floor context, plasticity refers to the collective ability of the robots to switch from doing no-task option (random-walking) to doing a task (or vice-versa) depending on the work-load present in the system. Here we expect to see that most of the robots would be able to engage in tasks when there would be high workloads (or task-urgencies) during PMM. Similarity, when there would be low workload in case of MOM, only a few robots would do the task, rest of them would either be idle (not doing any task) or perform a random-walk. The changes of task-urgencies and the ratio of robots engaged in tasks can be good metrics to observe plasticity in MRTA.

Task-specialization: Self-regulated MRTA is generally accompanied with task-specializations of agents. That means that few robots will be more active than others. From the interpretation of AFM, we can see that after doing a task a few times, a robot will soon be sensitized to it. Therefore, from the raw log of task-sensitization of robots, we can be able to find the pattern of task-sensitization of robots per task basis.

Quality of task-performance: As discussed in Sec. **??** we can measure the quality of MRTA from the APCD. It first calculates the ideal minimum production time and then finds the delay in production process from the actual production completion data. Thus this will indicate how much more time is spent in the production process due to the self-regulation of robots in this distributed task-allocation scheme. In order to calculate APCD, we can find the production completion time for each task from the raw log of task-urgency and make an average from them.

Robustness: In order to see if our system can respond to the gradually increasing workloads, we can measure APMW within the context of our manufacturing shop-floor scenario. This can show the robustness of our system. When a task is not being served by any robot for some time we can see that its urgency will rise and robots will respond to this dynamic demand. For measuring APMW we need only the task-urgency data.

Table 1 Experimental parameters of Series A & B experiments

Parameter	Series A | Series B
Total number of robots (N)	8 | 16
Total number of tasks (M)	2 | 4
Experiment area (A)	2 m^2 | 4 m^2
Initial production work load/machine (Ω_j^p)	100 unit
Task urgency increase rate ($\Delta\phi_{INC}$)	0.005
Task urgency decrease rate ($\Delta\phi_{DEC}$)	0.0025
Initial sensitization (K_{INIT})	0.1
Sensitization increase rate (Δk_{INC})	0.03
Sensitization decrease rate (Δk_{DEC})	0.01

Flexibility: From the design of AFM, we know that robots that are not doing a task will be de-sensitized to it or forget that task. So at an overall low work-load (or task urgency), less robots will do the tasks and hence less robots will have the opportunity to learn tasks. From the shop-floor work-load data, we can confirm the presence of flexibility in MRTA.

Energy-efficiency: In order to characterize the energy-efficiency in MRTA we can log the pose data of each robot that can give us the total translations occurred by all robots in our experiments. This can give us a rough indication of energy-usage by our robots.

Information flow: Since AFM requires a system-wide continuous flow of information, we can measure the communication load to bench-mark our implementation of communication system. This bench-mark data can be used to compare among various communication strategies. Here we can measure how much task-related information, i.e. task-urgency, location etc. are sent to the robots at each time step. This amount of information or communication load can be constant or variable depending on the design of the communication system.

Scalability: In order to see the effects of scaling on MRTA, we have designed two group of experiments. Series A corresponds to a small group where we have used 8 robots, 2 tasks under an arena of 2 m^2. We have doubled these numbers in Series B, C and D, i.e. 16 robots, 4 tasks under an arena of 4 m^2. This proportional design can give us a valuable insight about the effects of scaling on self-regulated MRTA.

Thus, in order to observe the above properties of self-regulated MRTA , we have designed our experiments to record the following observables in each time-step.

1. Task-urgency of each task (ϕ).
2. Number of robots engaged in each task.
3. Task-sensitizations (k) of robots.
4. Pose data of robots.
5. Communication of task-information message among TPS and RCCs.

4.2 Parameters

Table 1 lists a set of essential parameters of our GSNC startegy based experiments (Series A and B). We intend to have a set-up that is relatively complex, i.e., with a high number of robots and tasks in a large area. The diameter of the marker of our e-puck robot is 0.08m

(Fig **??**). So, if we put 4 robots in an area of one square meter, this will give us a robot-occupied-space to free-space ratio of about 1:49 per square meter. This ratio is reasonable in order to allow the robots to move at a speed of 5 cm/sec without causing much interference to each other.

The initial values of task urgencies correspond to 100 units of production work-load without any maintenance work-load as outlined in Eq. 8. We choose a limit of 0 and 1, where 0 means no urgency and 1 means maximum urgency. Same rule applies to sensitisation, where 0 means no sensitisation and 1 means maximum sensitisation. This also implies that if sensitization is 0, task has been forgotten completely. On the other hand, if sensitization is 1, the task has been learnt completely. We choose a initial sensitization value of 0.1 for all tasks. The following relationships are maintained for selecting task-urgency and sensitization parameters.

$$\Delta\phi_{INC} = \frac{\Delta\phi_{DEC} \times N}{2 \times M} \tag{17}$$

$$\Delta k_{DEC} = \frac{\Delta k_{INC}}{M-1} \tag{18}$$

Eq. 17 establishes the fact that task urgency will increase at a higher rate than that of its decrease. As we do not like to keep a task left unattended for a long time we choose a higher rate of increase of task urgency. This difference is set on the basis of our assumption that at least half of the expected number of robots (ratio of number of robots to tasks) would be available to work on a task. So they would produce similar types of increase and decrease behaviours in task urgencies.

Eq. 18 suggests that the learning will happen much faster than the forgetting. The difference in these two rates is based on the fact that faster leaning gives a robot more chances to select a task in next time-step and thus it becomes more specialized on it.

4.3 Implementation

We have developed a system a multi-robot tracking system can track at least 40 E-puck robots [1] and these robots can operate together according to the generic rules of the AFM. As shown in Fig. 4, our software system consists of a multi-robot tracking system, a centralized task server and robot controller clients. Here at first we have presented the design of our communication system. Then we have discussed about our specific implementation. In order to establish a system-wide continuous flow of information, we need to implement a suitable communication system for our robots. Here we have presented a centralized communication system for our manufacturing shop-floor scenario. As shown in Fig. 3, in this model there exists a centralized *TaskServer* that is responsible for disseminating task information to robots. The contents of task information can be physical locations of tasks, their urgencies and so on. TaskServer delivers this information by emitting *TaskInfo* signals periodically. The method of signal emission depends on a particular communication technology. For example, in a wireless network it can be a message broadcast. Task-Server has another interface for catching feedback signals from robots. The *RobotStatus* signal can be used to inform TaskServer about a robot's current task id, its device status and so on. TaskServer uses this information to update relevant part of task information such as, task-urgency. This up-to-date information is sent in next TaskInfo signal.
In Fig. 3 an initial configuration of this model has been presented. Upon receiving an initial

[1] www.e-puck.org

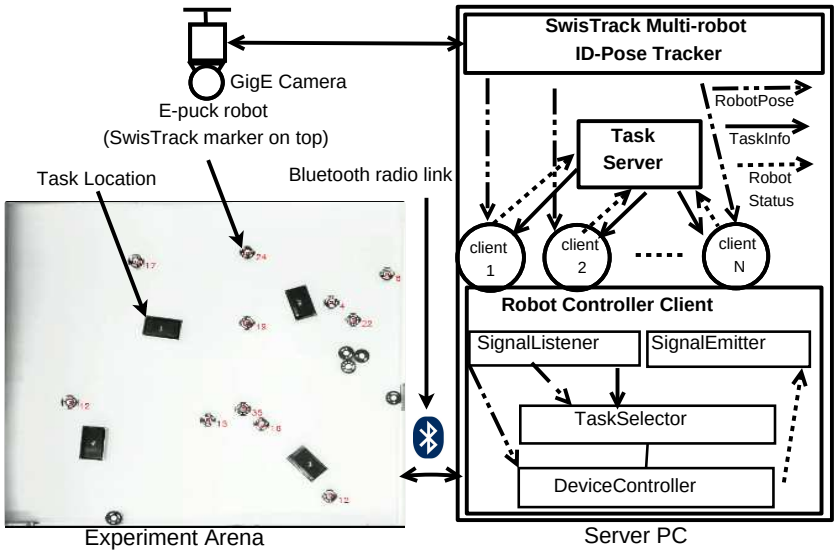

Fig. 4 Hardware and software setup

TaskInfo signal robot R_1 has shown strong attraction towards *Task*1 and robot R_3 has shown strong attraction toward *Task*2. This can be inferred from Eq. **??** that says if the initial task urgencies and sensitizations for all tasks are same, a robot will strongly be attracted towards a task that is relatively closer to it. The major components of our implementation are a multi-robot tracking system, robot controller clients and a centralized task-server. In order to track all robots real-time we have used SwisTrack (Lochmatter et al, 2008), a state of the art open-source, multi-agent tracking system, with a16-megapixel overhead GigE camera. This set-up gives us the position, heading and id of each of the robots at a frequency of 1. The interaction of the hardware and software of our system is illustrated in Fig. 4.

For inter-process communication (IPC), we have used D-Bus technology [2]. We have developed an IPC component for SwisTrack (hereafter called as *SwisTrack D-Bus Server*) that can broadcast id and pose of all robots in real-time over our server's D-Bus interface.

Apart from SwisTrack, we have implemented two major software modules: *TaskServer* and *Robot Controller Client (RCC)*. They are developed in Python with its state of the art *Multiprocessing* [3] module. This python module simplifies our need to manage data sharing and synchronization among different sub-processes. As shown in Fig. 4, RCC consists of four sub-processes. *SignalListener* and *SignalEmitter*, interface with SwisTrack D-Bus Server and TaskServer respectively. *TaskSelector* implements AFM guidelines for task selection . *DeviceController* moves a robot to a target task. Bluetooth radio link is used as a communication medium between a RCC and a corresponding E-puck robot.

[2] http://dbus.freedesktop.org/doc/dbus-specification.html
[3] http://docs.python.org/library/multiprocessing.html

5 Results

In this section we have presented our experimental results. We ran those experiments for about 40 minutes and averaged them over five iterations for both Series A and B.

Shop-floor work-load history

In our experiments we have defined shop-floor work-load in terms of task urgencies. For example, Eq. 8 shows how we have calculated initial production work-load of our manufacturing shop-floor scenario. Fig. 5 and Fig. 6 show the dynamic changes in task-urgencies for the single iteration of Series A and Series B experiments respectively. The fluctuations in these plots are resulted from the different levels of task-performance of our robots. In order to measure the task-related work-loads on our system we have summed up the changes in all task-urgencies over time. We call this as *shop-floor work-load history* and formalized as follows. Let $\phi_{j,q}$ be the urgency of a task j at q^{th} step and $\phi_{j,q+1}$ be the task urgency of $(q+1)^{th}$ step. We can calculate the sum of changes in urgencies of all M tasks at $(q+1)^{th}$ step:

$$\Delta\Phi_{j,q+1} = \sum_{j=1}^{M} (\phi_{j,q+1} - \phi_{j,q}) \tag{19}$$

From Fig. 7 and Fig. 8 show the dynamic shop-floor workload for Series A and Series B experiments respectively. From these plots, we can see that initially the sum of changes of task urgencies (shop-floor workload) is going towards negative direction. This implies that tasks are being served by a high number of robots.

Ratio of active workers

From both Fig. 9 and Fig. 10, we can see that in production stage, when work-load is high, many robots are active in tasks. Here active workers ratio is the ratio of those robots that work on tasks to the total number of robots N of a particular experiment. Here we can see that this ratio varies according to the shop-floor work-load changes.

Shop-task performance

In our manufacturing shop-floor scenario, we have calculated the APCD
and APMW for both Series A and Series B experiments. For Series A we have got average production completion time 111 time-steps (555s) where sample size is (5 x 2) = 10 tasks, SD = 10 time-steps (50s). According to Eq. 12, our theoretical minimum production completion time is 50 time-steps (250s) assuming the non-stop task performance of all 8 robots with an initial task urgency of 0.5 for all 2 tasks and task urgency decrease rate $\Delta\Phi_{DEC}$ = 0.0025 per robot per time-step. Hence, Eq. 13 gives us APCD, ζ = 1.22 which means that in Series A experiments, it took 1.22 times more time (305s) than the estimated minimum production completion time (250s). For Series B, we have got average production completion time 165 time-steps (825s) where sample size is (5 x 4) = 20 tasks, SD = 72 time-steps (360s). Hence, Eq. 13 gives us APCD, ζ = 2.3. Fig. **??** shows the APCD for both Series A and Series B experiments.

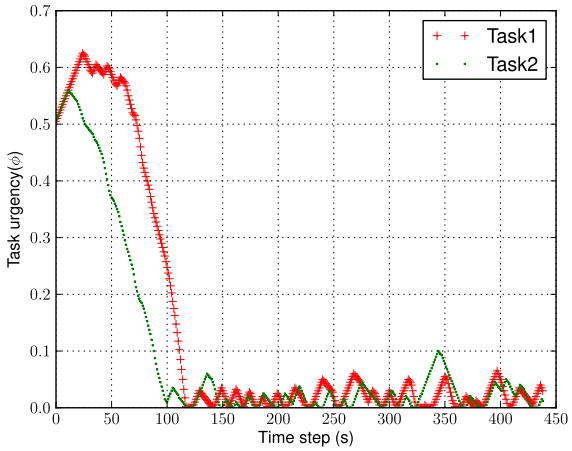

Fig. 5 Changes in task-urgencies in Series A experiments

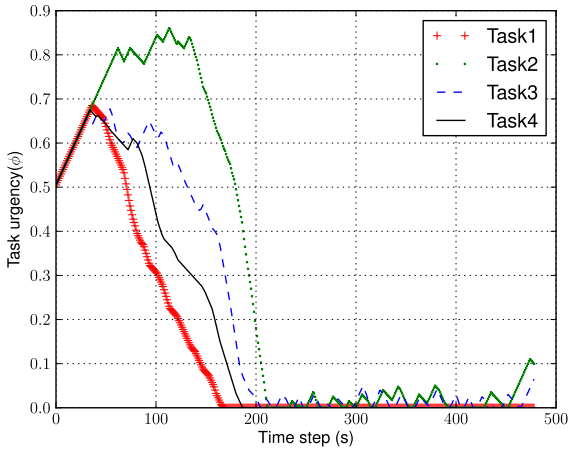

Fig. 6 Changes in task-urgencies in Series B experiments

For APMW, Series A experiments give us an average time length of 369 time-steps (1845s). In this period we calculated APMW and it is 1 time-step with SD = 1 time-step (5s) and $\Delta\Phi_{INC}$ = 0.005 per task per time-step. This shows a very low APMW (χ = 0.000235) and a very high robustness of the system. For Series B experiments, from the average 315 time-steps (1575s) maintenance activity of our robots per experiment run, we have got APMW, χ = 0.012756 which corresponds to the pending work of 3 time-steps (15s) where SD = 13 time-steps (65s). This tells us the robust task performance of our robots which can return to an abandoned task within a minute or so. Fig. **??** plots the APMW for both Series A and Series B experiments.

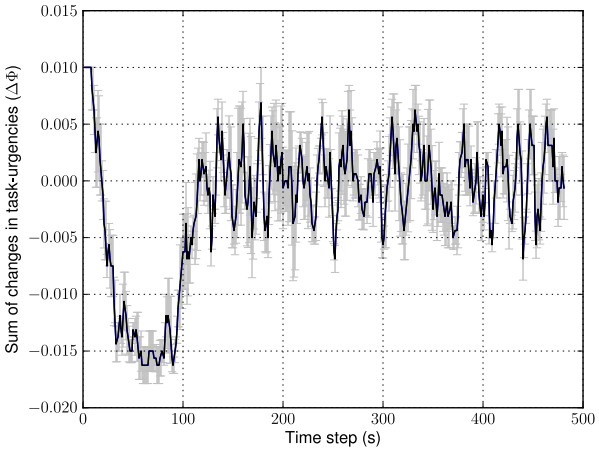

Fig. 7 Shop-floor workload change history in Series A

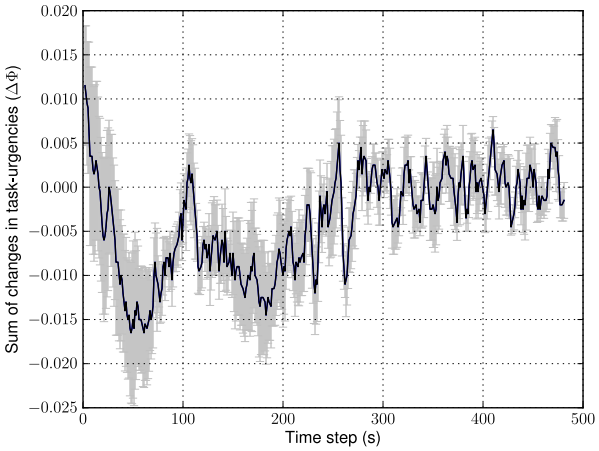

Fig. 8 Shop-floor workload change history in Series B

Task specializations

We have measured the task-specialization of the robots based-on their peak value of sensitization. This maximum value represents how long a robot has repeatedly been selecting a particular task. Since tasks are homogeneous we have considered the maximum sensitization value of a robot among all tasks during an experiment run. This value is then averaged for all robots using the following equation.

$$K_{avg}^{G} = \frac{1}{N}\sum_{i=1}^{N}\max_{j=1}^{M}\left(k_{j,q}^{i}\right) \tag{20}$$

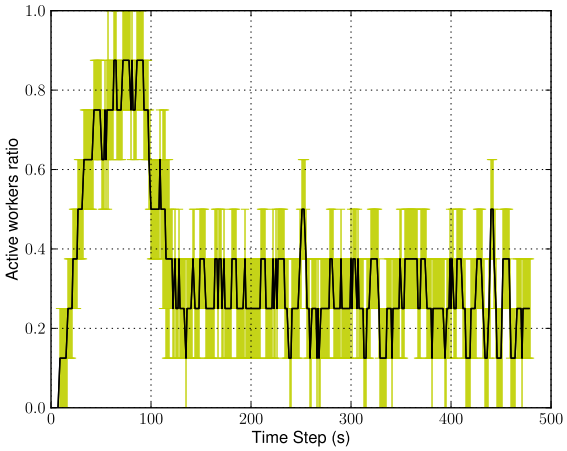

Fig. 9 Self-organized allocation of robots in Series A

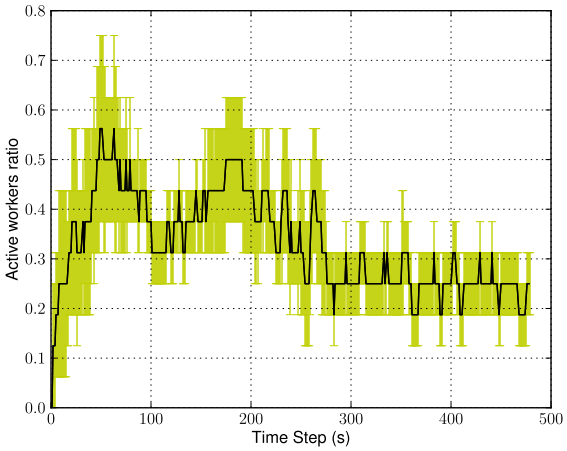

Fig. 10 Self-organized allocation of robots in Series B

If a robot r_i has the peak sensitization value k_j^i on task j ($j \in M$ tasks) at q^{th} time-step, Eq. 20 calculates the average of the peak task-specialization values of all robots for a certain iteration of our experiments. We have also averaged the time-step values (q) taken to reach those peak values for all robots using the following equation.

$$Q_{avg}^G = \frac{1}{N} \sum_{i=1}^{N} q_{k=k_{max}}^i \tag{21}$$

In Eq. 21, $q_{k=k_{max}}^i$ represents the time-step of robot r_i where its sensitization value k reaches the peak k_{max} as discussed above. By averaging this peak time-step values of all robots we can have an overall idea of how many task-execution cycles are spent to reach the maximum task-specialization value K_{avg}^G. Table 2 and Table 3 show the peak sensitization values of

Table 2 Peak task-sensitization values of robots in a Series A experiment.

Robot ID	Maximum k	At time-step (q)	Task
1	0.54	64	Task1
4	0.32	14	,,
5	0.27	11	,,
3	0.47	63	Task2
2	0.46	64	,,
6	0.20	10	,,
7	0.18	4	,,
8	0.15	3	,,

Table 3 Peak task-sensitization values of robots in a Series B experiment.

Robot ID	Maximum k	At time-step (q)	Task
24	0.41	29	Task1
13	0.31	19	,,
16	0.18	4	,,
1	0.64	66	Task2
35	0.34	12	,,
5	0.28	14	,,
22	0.18	20	,,
17	0.16	6	,,
3	0.14	12	,,
9	0.68	66	Task3
6	0.43	71	,,
15	0.19	4	,,
14	0.15	4	,,
31	0.14	4	,,
19	0.22	12	Task4
12	0.16	10	,,

Series A and Series B experiments respectively. Based on Eq. 20 and Eq. 21, we have got the peak task-sensitization K^G_{avg} values: 0.40 (SD=0.08) and 0.30 (SD=0.03), and their respective time-step Q^G_{avg} values: 38 (SD=13) and 18 (SD=5) time-step. They are shown in Fig. **??** and Fig. **??**. Here we can see that the robots in Series A had higher chances of task-specialization than that of Series B experiments. Fig. 11 shows us the task specialization of five robots on Task3 in a particular run of Series B experiment. This shows us how some of the robots can specialize (learn) and de-specialize (forget) tasks over time.

Robot motions

We have aggregated the changes in translation motion of all robots over time. Let $u_{i,q}$ and $u_{i,q+1}$ be the translations of a robot i in two consecutive steps. If the difference between these two translations be δu_i, we can find the sum of changes of translations of all robots in

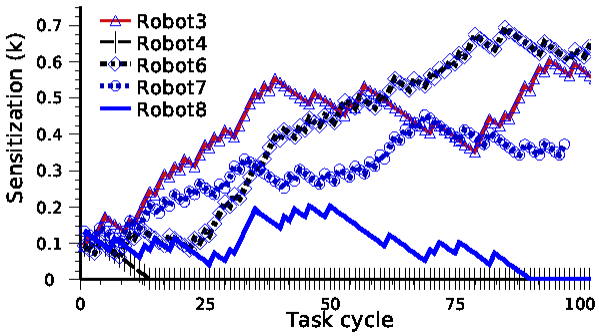

Fig. 11 Task specialization on Task3 for a Series B experiment.

$(q+1)^{th}$ step using the following equation.

$$\Delta U_{q+1} = \sum_{i=1}^{N} \delta u_{i,q+1} \tag{22}$$

The results from Series A and Series B experiments are plotted in Fig. 12 and Fig. 13. In this plot we can see that robot translations also vary over varying task requirements of tasks.

Communication load

Fig. 14 and Fig. 15 show the number of received `TaskInfo` signals by each robot in Series A and Series B experiments. Since the duration of each time-step is 50s long and TPS emits signal in every 2.5s, there is an average of 20 signals in each time-step.

6 Discussions

Self-regulated DOL. From our experimental results, we have noted several aspects of self-regulated DOL that exposes the power of AFM. As we have pointed out that this self-regulated DOL, as observed in biological and human social systems, needs to satisfy several important characteristics, e.g. plasticity, task-specialization. In addition to satisfying those basic qualities, AFM has demonstrated many other aspects. Our self-regulated robots, driven by AFM, effectively handle the dynamic work-load in our manufacturing shop-floor. They can dynamically support the need to work on currently demanding tasks, if there any. The variations of active worker ratio supports this.

From the self-organized worker allocations of AFM, it is clear to us that although in larger system (Series B) the degree of variations of active-worker ratio can show us significantly unpredictable patterns, nevertheless the self-regulated rules drive the robots to respond to the dynamic needs of the system. This means that AFM can sufficiently produce the plasticity of DOL in order to meets the dynamic work-load of the system.

Learning and Forgetting. From the individual and group-level task-specialization, we can see that robots can maintain both task-specialization and flexibility.

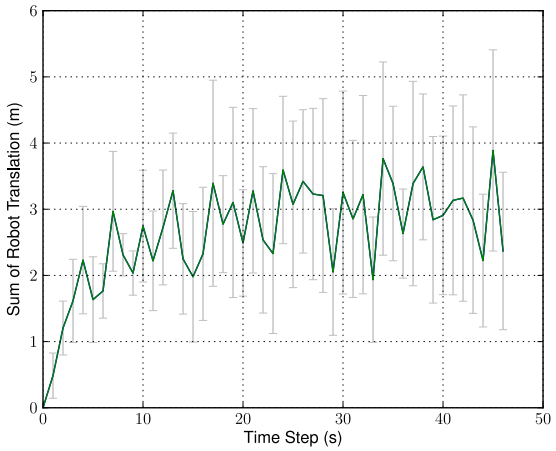

Fig. 12 Sum of the translations of robots in Series A experiments

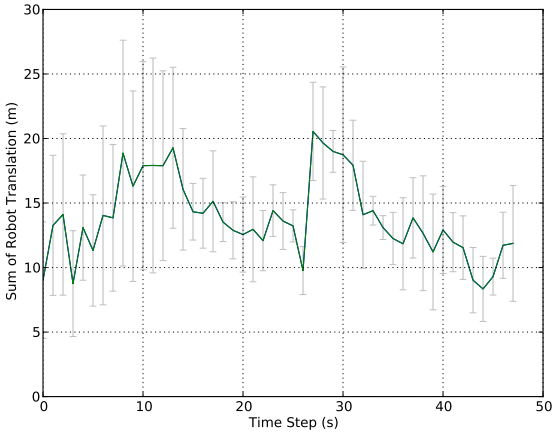

Fig. 13 Sum of the translations of robots in Series B experiments

In a self-organized system, it is very common that only a few individuals specialize on tasks and others generally do not. From two samples data sets, we can see that in particular runs of Series A and Series B experiments, task-sensitization values of only 2-3 robots reach above the group-level average score. Thus in both types of experiments, robots exhibit similar task-specialization behaviours.

From task-sensitization we can also see that a limited number of robots are specialized in tasks. Thus most of the other robots are flexible in selecting any tasks as their task-specializations do not bind them to particular tasks.

Concurrency and robustness. As a consequence of fewer robots specializing in tasks, we can also see that robots can concurrently consider different tasks without being biased to a particular task all the time. Our experiments also show us the robust DOL as in case of both high and low work-loads present in the system. This is evident from the manufacturing

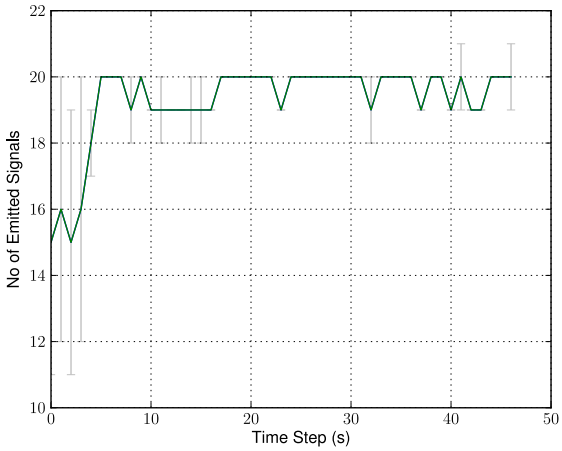

Fig. 14 Frequency of `TaskInfo` signalling in Series A experiments

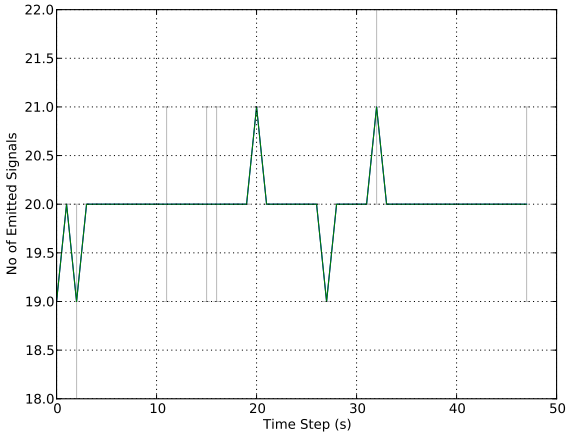

Fig. 15 Frequency of `TaskInfo` signalling in Series B experiments

shop-floor task performance during PMM and MOM. For example, in case of Series B experiments APMW was 13 time-steps (65s) which corresponded to pending work-load of 0.065 unit for a single robot. Thus, on an average, before the work-load exceeded by about 13 percent of initial work-load, robots were able to respond to a task.

Communication load. In these experiments we used a centralized communication system (source of attractive fields) that serves the robot with necessary task-perception information. Although our robot-controllers software RCC was also co-located in the same host-PC, they can be distributed to several PCs or robot's on-board PCs. Our centralized communication system has the advantage of minimising the communication load and the disadvantage of a single point of failure as well as single point of load. In the next chapter we present how the task perception can be decentralized by P2P communications among RCCs.

Scaling-up. We have observed the effect of scaling-up the robot team size. The system size of Series B is double of that of Series A in terms of robots, tasks and experiment arena. Keeping a fixed ratio of robot-to-task and task-to-arena we have intended to see the scaling effects in our experiments. Here we see that both systems can show sufficient self-regulated DOL, but task-performance of both systems varies significantly. For example, the value of APCD in Series B is higher by 1.08. This means that performance is decreased in Series B experiments despite having the resources in same proportion in both systems. This occurs partly due to the greater stochastic effects found in task-allocation in a larger system, e.g. presence of more tasks produce higher stochastic behaviours in robot's task selection.

Similarly we can see that in larger system robots have less chances to specialize on tasks, as the Series B experiments show us that the overall average task-specialization of the group K^G_{avg} is lower by 0.10 and it lasts for significantly less time (the difference of Q^G_{avg} of both systems is 20 time-steps). Thus, in a large group, robots are more likely to switch among tasks more frequently and this produces more translation motions which cost more energy (e.g. battery power) in task-performance.

7 Related Work

Traditionally MRTA solutions are classified into two broad categories: i) *explicit* task-allocation and 2) *bio-inspired self-organized* task-allocation. Explicit approach allocates task through explicit modelling of environment, tasks, robot capabilities. Some forms are: knowledge based (Parker, 1998), market based (Dias et al, 2006), role/value based Chaimowicz et al (2002), control theoretic Belta and Kumar (2004). This approach is straight-forward to design, implement and analyse formally. But they are not suitable for large teams (> 10) and heavily dependent upon explicit global broadcast communication. On the other hand self-organized approach allocates tasks through emergent group behaviour produced by the local interaction and implicit or local communication Bonabeau et al (1999). This approach is suitable for large teams since it has no explicit model and uses implicit/local communication. But this approach is difficult to design, implement, analyse and limited to one specific global task (Gerkey and Mataric, 2004).

Both of these approaches expose their relative strengths and weaknesses when they are put under real-time experiments with variable number of robots and tasks. In an arbitrary event handling domain, a comparison between self-organized and predefined market-based task-allocation was reported by Kalra and Martinoli Kalra and Martinoli (2007). They found that explicit task-allocation was more efficient when the required information could be captured accurately. The threshold-based approach offered similar quality of allocation at a fraction of cost in noisy environment.

Task performance in self-organized approaches relies on the collective behaviours resulting from the local interactions of many simple and mostly homogeneous or interchangeable agents. Robots choose their tasks independently using general principles of self-organization such as: positive and negative feedback, multiple interactions among entities and their environment, and randomness or amplification of fluctuations Camazine et al (2001). Moreover. interaction among individuals and their environment can be modulated by stigmergic, local and global communications. Among many variants of self-organized task-allocation mechanisms, the most common type is threshold-based task-allocation Bonabeau et al (1999). In this approach, a robot's decision to select a particular task depends largely on its perception of a stimulus, e.g., the demand for a task to be performed, and its corresponding response threshold for that task.

Under the deterministic response-threshold approach, each robot has an activation threshold for each task that needs to be performed. It continuously perceives or monitors the stimulus of all tasks that reflect the relative urgencies of tasks. When a particular task-stimulus exceeds a predefined threshold the robot starts working on that task, typically reducing the related stimuli. When the task-stimuli falls below the fixed threshold the robot abandons that task. This type of approach has been effectively applied in foraging Liu et al (2007); Krieger and Billeter (2000) and aggregation Agassounon and Martinoli (2002). The fixed response threshold can initially be same for all robots Jones and Mataric (2003) or they can be different according robot capabilities or the configuration of the system Krieger and Billeter (2000). Adaptive response-threshold models change the thresholds over time. A robot's response threshold for a given task is often decreased as a result of the robot performing that task. This enables a robot to select that particular task more frequently or in other words specialise on that task Bonabeau et al (1999); Agassounon and Martinoli (2002). Unlike the deterministic approach, where robots respond predictably, e.g., to the task stimulus that is the furthest above its threshold, probabilistic or stochastic approaches offer a selection process based-on a probability distribution over the tasks. In this case, all tasks commonly have at least a small, non-zero probability of being chosen. This random element typically prevents starvation of low stimulus tasks.

8 Conclusions and Future works

In this paper we have validated an inter-disciplinary generic model of self-regulated division of labour (DOL) or or multi-robot task allocation (MRTA) by incorporating it in our multi-robot system (MRS) that has emulated a virtual manufacturing shop-floor activities. A centralized communication system has been instantiated to realize this model. We have evaluated various aspects of this model, such as ability to meet dynamic task demands, individual task specializations, communication loads and flexibility in concurrent task completions. A set of metrics has been proposed to observe the DOL in this system. From our experimental results, we have found that AFM can meet the requirements of dynamic DOL by the virtue of its self-regulatory behaviours. Our centralised communication system broadcasts information to all the robots from a central server. This has the advantage of minimising the communication load and the disadvantage of a single point of failure. In the future, we will explore local peer-to-peer communication models in a MRS having about 40 E-puck robots.

Acknowledgements.

This research has been funded by the Engineering and Physical Sciences Research Council (EPSRC), UK, grant reference EP/E061915/1.

References

Agassounon W, Martinoli A (2002) Efficiency and robustness of threshold-based distributed allocation algorithms in multi-agent systems. In: Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 3, ACM, p 1097

Arcaute E, Christensen K, Sendova-Franks A, Dahl T, Espinosa A, Jensen HJ (2008) Division of labour in ant colonies in terms of attractive fields. Ecol Complexity

Belta C, Kumar V (2004) Abstraction and control for groups of robots. IEEE Transactions on Robotics 20(5):865–875

Bonabeau E, Dorigo M, Theraulaz G (1999) Swarm intelligence: from natural to artificial systems. Oxford University Press

Camazine S, Franks N, Sneyd J, Bonabeau E, Deneubourg J, Theraula G (2001) Self-organization in biological systems. Princeton, N.J. : Princeton University Press, c2001., what is self-organization? – How self-organization works – Characteristics of self-organizing systems – Alternatives to self-organization –

Chaimowicz L, Campos MFM, Kumar V (2002) Dynamic role assignment for cooperative robots. Robotics and Automation, 2002 Proceedings ICRA'02 IEEE International Conference on 1

Dias MB, Zlot RM, Kalra N, Stentz A (2006) Market-based multirobot coordination: A survey and analysis. Proceedings of the IEEE 94:1257–1270

Gerkey B, Mataric M (2001) Principled communication for dynamic multi-robot task allocation. Experimental Robotics VII pp 353–362

Gerkey BP, Mataric MJ (2004) A formal analysis and taxonomy of task allocation in multi-robot systems. The International Journal of Robotics Research 23:939

Jones C, Mataric M (2003) Adaptive division of labor in large-scale minimalist multi-robot systems. In: 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings

Kalra N, Martinoli A (2007) A comparative study of market-based and threshold-based task allocation. Distributed Autonomous Robotic Systems 7 pp 91–101

Krieger M, Billeter J (2000) The call of duty: Self-organised task allocation in a population of up to twelve mobile robots. Robotics and Autonomous Systems 30(1-2):65–84

Kube CR, Zhang H (1993) Collective robotics: From social insects to robots. Adaptive Behavior 2:189

Lerman K, Jones C, Galstyan A, Mataric MJ (2006) Analysis of dynamic task allocation in multi-robot systems. The International Journal of Robotics Research 25:225

Liu W, Winfield AFT, Sa J, Chen J, Dou L (2007) Towards energy optimization: Emergent task allocation in a swarm of foraging robots. Adaptive Behavior 15(3):289–305

Lochmatter T, Roduit P, Cianci C, Correll N, Jacot J, Martinoli A (2008) Swistrack-a flexible open source tracking software for multi-agent systems. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008. IROS 2008, pp 4004–4010

Parker LE (1998) Alliance: an architecture for fault tolerant multirobot cooperation. Robotics and Automation, IEEE Transactions on 14:220–240

Parker LE (2008) Distributed intelligence: Overview of the field and its application in multi-robot systems. Journal of Physical Agents, special issue on multi-robot systems vol. 2(no. 2):5–14