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

A Robotic Validation of the Attractive Field Model: An Inter-Disciplinary Model of Self-Regulatory Social Systems

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

the date of receipt and acceptance should be inserted later

Abstract Division of labour in multi-robot systems or multi-robot task allocation (MRTA) is a challenging research issue. We propose to solve this MRTA problem using a set of previously published generic rules for division of labour derived from the observation of ant, human and robotic social systems. These bottom-up rules describe the phenomenon of self-regulated division of labour in terms of attractive fields between robots and tasks. The concrete form of these rules, the *attractive filed model*, avoids the strong dependence on local interactions found in many existing approaches to MRTA. We present experimental results that constitute a first validation of attractive filed model as a mechanism for MRTA and as a multi-disciplinary model of self-organisation in social systems. Our experiments used upto 16 e-puck robots in a 2m x 2m area.

1 Introduction

Scientific studies show that a large number of animal as well as human social systems grow, evolve and generally continue functioning well by the virtue of their individual self-regulatory mechanism of division of labour (DOL) (Bonabeau et al, 1999). This has been accomplished without any central authority or any explicit planning and coordinating element. Indirect communication such as stigmergy is instead used to exchange information among individuals. In robotic systems, *multi-robot task allocation* (MRTA) is a common research challenge (Gerkey and Mataric, 2004). It is generally identified as the problem of assigning tasks to appropriate robots at appropriate times taking into account potential changes in the environment and/or the performance of the robots. MRTA is an optimal assignment problem that has been shown to be NP-hard, so optimal solutions can not be expected for large problems (Parker, 2008). In addition to the inherent complexity of MRTA, the problem is also commonly restricted to avoid central planners or coordinators for task assignments. The robots are also commonly limited to local sensing, communication and interaction (Lerman et al, 2006) where no single robot has complete knowledge of the past, present or future actions of other robots or a complete view of the world state. For larger teams of robots the bandwidth of local communication channels is also limited. In practical implementations

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

the computational and communication bandwidth requirements restrict the quality of the solutions to MRTA problems (Gerkey and Mataric, 2004; Lerman et al, 2006).
Traditionally MRTA solutions are divided into two major categories: 1) Predefined (off-line) and 2) Emergent (real-time) task-allocation (Shen et al, 2001). Early research on predefined task-allocation approaches was dominated by intentional coordination, use of dynamic role assignment (Parker, 2008) and market-based bidding approach (Dias et al, 2006). In the intentional approaches the robots use direct task-allocation method to communicate and to negotiate tasks. This approach is intuitive, comparatively straightforward to design and implement and can be analysed formally. However, this approach typically works well only when the number of robots is small (Lerman et al, 2006).The emergent task-allocation approach on the other hand, relies on the emergence of group behaviours, e.g., emergent cooperation (Lerman et al, 2006), using mechanisms such as *adaptation rules* (Liu et al, 2007). This approach typically handles systems with local sensing, local interactions and typically little or no explicit communication or negotiations between robots. Emergent systems are more scalable and robust due to their inherent parallelism and redundancy. However in these systems, solutions are unintuitive and thus difficult to design, analyse formally and implement practically (Gerkey and Mataric, 2004; Lerman et al, 2006). The solutions found by these systems are typically sub-optimal and, as the emergence is a result of interactions among robots and their environment, it is also difficult to predict exact behaviours of robots and overall system performance. The current challenges in emergent task allocation approaches have lead us to look for a suitable alternative. In nature we find that task allocation in animal or insect societies can be governed by non-centralized rules and that they are self-regulating and self-stabilizing (Bonabeau et al, 1999). Moreover studies in sociology (Sayer and Walker, 1992), strategic management (Kogut, 2000) and related disciplines show that decentralized self-regulated systems exist and that they can survive and grow over time.
As a part of a collaborative project, we have studied the behaviour of ants, humans and robots and have developed the attractive field model (AFM), a common formal model of division of labour in social systems (Arcaute et al, 2008). In this paper, we present an application of AFM in a robotic system. Section 2 presents AFM and an associated communication model that allows us to implement AFM as a MTRA mechanism. Section 4 introduces our implementation of MRTA including the interactions between the hardware, software and communication modules. Section 5 presents the design of our experiments including specific parameters and observables. Section 6 discusses our experimental results and section 7 draws conclusions.

2 The Attractive Field Model

The AFM has been developed through a collaborative between the partners on the 'Defying the Rules: How Self-Regulated Social; Systems Work' project. Experiments on ants colonies *Temnothorax albipennis* studied 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. We also developed a concrete model of self-organisation based on task-allocation. This model was validated by deploying it as a robot controller mechanism. In this section, we describe the generic AFM and how it embodies the requirements for self-organization. We also provide an interpretation of AFM in multi-robot systems.

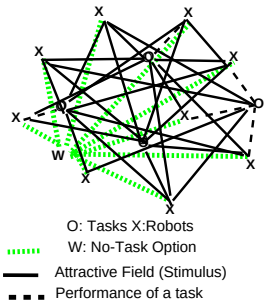

Fig. 1 Attractive Filed Model (AFM)

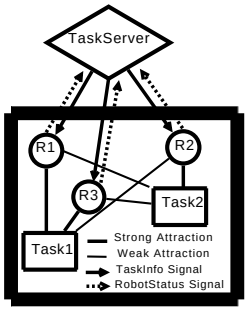

Fig. 2 A centralized communication scheme

2.1 The Attractive Field Model

Inspired from the DOL in ants, humans and robots, we have proposed the following necessary and sufficient set of four requirements for self-regulation in social systems.

Requirement 1: Concurrence The simultaneous presence of several options 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.

Requirement 2: Continuous flow of information Self-organised social systems 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 agents self-organise by enabling them to perceive tasks and receive feedback on system performance.

Requirement 3: Sensitization The system must have a way of representing the structure produced by self-organisation, in terms of MRTA, which tasks the robots are allocated. One of the simplest ways of representing this information is an individual preference parameter for each task-robot combination. A system where each robot has different levels of preference or *sensitivity* to the available tasks, can be said to have to embody a distinct organisation through differentiation.

Requirement 4: Forgetting When a system self-organises by repeated increases in individual sensitisation levels, it is also necessary, in order to avoid saturation, to have a mechanism by which the sensitisation levels are reduced or *forgotten*. In addition to avoiding the situation where a structure produced by self-organisation is eroded by an eventual increase of sensitisation values to a given maximum, forgetting also allows flexibility in the system, in that the structure can change as certain tasks become important and other tasks become less so. This effect can be achieved by mechanisms such as a slow general decay of sensitisation values or explicit negative feedback. 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. The model 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
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.

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

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

2.2 A Robotic Interpretation of AFM

The interpretation of AFM in a multi-robot system follows the above mentioned generic interpretation. Each robot is modelled as an agent and each task is modellled as a spatial location. The robots repeatedly select tasks and if the robot is outside a fixed task boundry, it navigates towards the task. If the robot is within the task boundry it remains there until the end of the time step when a new (or the same) task is selected. 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.

2.3 AFM and Biological Self-Organization

Fig. **??** labels the four distinct perspectives known as so-called ingredients or properties of self-organization: A) positive feedback, B) negative feedback, C) presence of multiple interactions among individuals and their environment, and D) amplification of fluctuations e.g., random walks, errors, random task-switching **?**. However, it is not clear how those properties can come into existence. Fig. **??** depicts the four underlying mechanisms (label 1 to 4) that explain how self-organization can be realized in different social systems using our generic framework. These are explained in the following paragraphs.

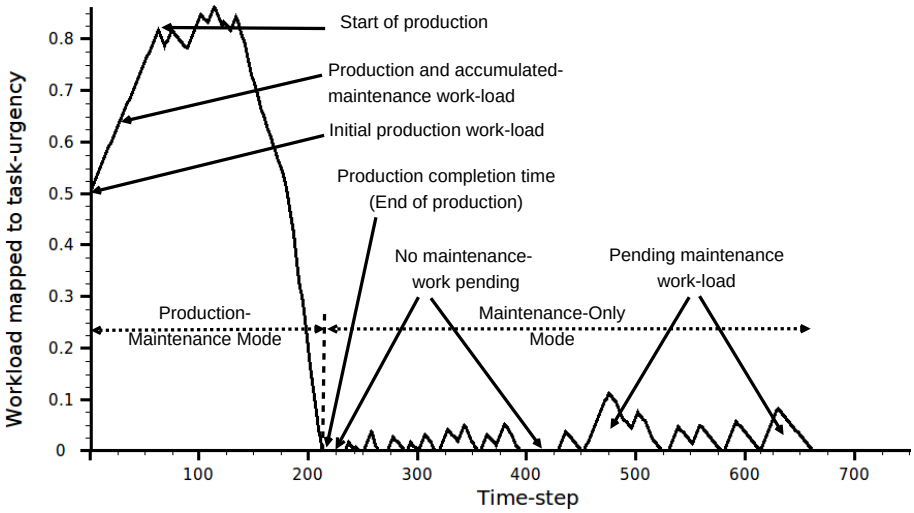

Fig. 3 Virtual Shop-floor production and maintenance cycle

Firstly, multiple interactions become meaningful when *continuous flow of information* occurs by exchanging signals or cues among agents or their environment that regulates their behaviours. This, in turn, contribute to the task-allocation and task-switching in the social level. In swarm intelligence literature, multiple interactions are often described as an essential ingredient of self-organization. However, interactions without definite purposes may not contribute to the self-organization.

Secondly, in swarm intelligence, positive feedback has been attributed as another mechanism of self-organization. But it is not easy to understand what creates positive feedback in a social system. Possible answers might be the characteristic of the environment e.g. ants select shorter path since density of pheromones becomes higher and thus more ants becomes attracted in that path, by causing the gradual decrease of response-threshold of individuals. This increases the probability of selecting a task. To make the answer more specific, we have explicitly attributed *sensitisation* or learning as a mechanism of positive feedback. There might exist other mechanisms too. But clearly sensitisation will be one of the reliable mechanisms for achieving positive feedback.

Thirdly, similar to positive feedback, we have proposed *forgetting* that contributes to provide negative feedback about a task or decreasing the probability to select it. Other negative feedback mechanisms can be implemented by assigning a saturation level to each task which is also present in our model, for details see Arcaute et al (2008).

Finally, creating artificial amplification of fluctuations or stochastic events is not a straightforward issue. It throws many open questions. Does a system designer intentionally impose irregularity in task-performance of agents? Is random movement enough for simulating randomness in a system? Since emergencies do not always pop-up on request, we provide the rule of *concurrency* that enables agents to maintain even a small amount of probability of selecting a low-priority, or less sensitized or distant task. This concurrency mechanism provides a high-degree of robustness in the system such that all tasks can be attended even if specialization of agents delays them in switching to some of the tasks.

2.4 A Manufacturing Shop-Floor Interpretation of AFM

We have designed a set of manufacturing shop-floor scenario experiments for validating the effectiveness of our 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 μ = 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. 3 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:

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

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).

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

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

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.

[1] www.e-puck.org

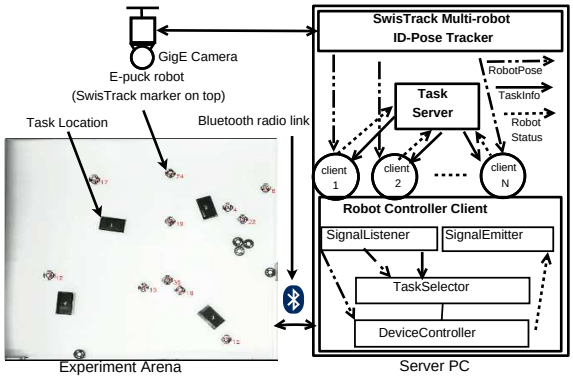

Fig. 4 Hardware and software setup

3.1 Design of our communication system

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. 2, 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. 2 an initial configuration of this model has been presented. Upon receiving an initial 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.

3.2 Our current implementation

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.

[2] http://dbus.freedesktop.org/doc/dbus-specification.html

Table 1 Experimental parameters

Parameter	Value
Total number of robots (N)	16
Total number of tasks (M)	4
Experiment area (A)	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

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.

4 Experiment Design

In this section, we have described the design of parameters and observables of our experiments within the context of our virtual 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 Parameters

Table 1 lists a set of essential parameters of our experiments. We intend to have a setup 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. 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 reasonable in order to allow the robots to move at a speed of 5 cm/sec without much interference to each other. We have fixed the number of tasks to 4. Robots also have an additional option for random walking that corresponds to the ignoring task information for ensuring flexibility of our system.
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 applies to sensitisation as well, 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

[3] http://docs.python.org/library/multiprocessing.html

is 1, the task has been learnt completely. We choose a default 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. Task-Server updates task-info messages in the interval of ΔTS_u=5s and robots stick on to a particular task for a maximum of ΔRT_{to}=10s.

4.2 Observables

We have defined a set of observables to evaluate our implementation. The first two observables, the changes in task-urgencies and the changes in active worker ratios, can give us an overall view of plasticity of division labour. Our third observable is to find changes in robot task specialization which is also an important measure of division of labour. Our last measurement is the communication load which is specific to this particular implementation and corresponds to the continuous flow of information. Within the context of our VMS, we measure the average production completion delay (APCD) and average pending maintenance work (APMW) as the metrics of VMS performance.

5 Results and Discussions

In this section we have presented our experimental results. We ran those experiments for about 40 minutes and averaged them over five iterations. Fig. 5 shows the dynamic changes in task urgencies. In order to describe our system's dynamic behaviour holistically we analyse the changes in task urgencies over time. 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 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. 6 we can see that initially the sum of changes of task urgencies are towards negative direction. This implies that tasks are being served by a high number of robots. Fig. 8 shows that in production stage, when work-load is high, many robots are active in tasks and this ratio varies according to task urgency changes.
Fig. 9 gives us the task specialization of five robots on Task3 in a particular run of our experiment. This shows us how our robots can specialize and de-specialize on tasks over time. The

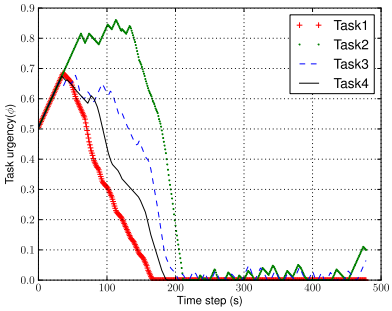

Fig. 5 Task urgencies observed at TaskServer

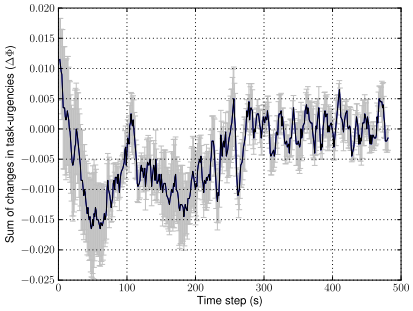

Fig. 6 Shop-floor workload change history

de-specialization or forgetting of tasks is calculated similar to Eq. 19. we have calculated the absolute sum of changes in sensitizations by all robots in the following equation.

$$\Delta K_{j,q+1} = \sum_{j=1}^{M} \left| (k_{j,q+1} - k_{j,q}) \right| \tag{20}$$

This values of ΔK are plotted in Fig. 10. It shows that the overall rate of learning decreases and forgetting increases over time. It is a consequence of the gradually increased task specialization of robots and reduced task-urgencies over time. Fig. 7 presents the frequency of signalling task information by TaskServer. Since the duration of each time step is 50s long and TaskServer emits signal in every 2.5s, there should be an average of 20 signals in each time-step.

Within VMS scenario, 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). According to Eq. 12, our theoretical minimum production completion time is 50 time-steps (250s) assuming the non-stop task performance of all 16 robots with an initial task urgency of 0.5 for all 4 tasks and task urgency decrease rate $\Delta\Phi_{DEC}$ = 0.0025 per robot per time-step. Hence, Eq. 13 gives us APCD, ζ = 2.3 which means that our system has taken 2.3 times more time (575s) than the minimum estimated time.

Besides, 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) with sample-size = 20 tasks, SD = 13 time-steps (65s), $\Delta\Phi_{INC}$ = 0.005 per task per time-step. This tells us the robust task performance of our robots which can return to an abandoned task within a minute or so.

6 Conclusion 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

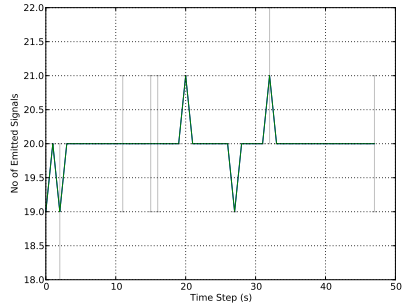

Fig. 7 Task server's frequency of task information signalling

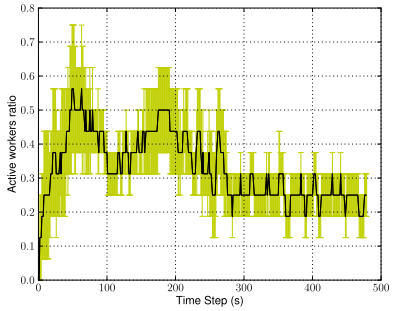

Fig. 8 Self-organized allocation of workers

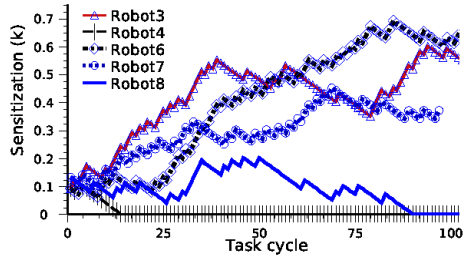

Fig. 9 Task specialization on Task3

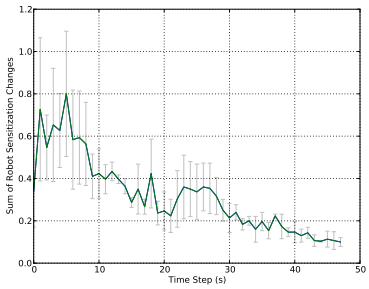

Fig. 10 Changes in sensitizations of all robots

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

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. In: Ecol. Complex

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

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

Kogut B (2000) The network as knowledge: generative rules and the emergence of structure. Strategic Management Journal 21(3):405–425

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 (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

Sayer A, Walker R (1992) The new social economy : reworking the division of labor. Blackwell, Oxford, 19920311

Shen W, Norrie DH, Barthes JP (2001) Multi-agent systems for concurrent intelligent design and manufacturing. Taylor & Francis, London