Poster in Proceedings of the Int. Conf. on Simulation of Adaptive Behavior (SAB), 395-396, Edinburgh, UK, Aug 2002

Sequential Task Execution in a Minimalist Distributed Robotic System

Chris Jones **Maja J. Matarić**
Computer Science Department
University of Southern California
941 West 37th Place, Mailcode 0781
Los Angeles, CA 90089-0781
{cvjones|maja}@robotics.usc.edu

1 Introduction and Related Work

A Minimalist Distributed Robotic System (MDRS) is a society of simple robots each with limited sensing, communication, and intelligence capabilities. We study ways of providing such MDRS with the capability of *sequential task execution, which requires a set of tasks to be executed in a specified order, with the initiation of a task occurring only after the termination of a required prior task.*. Sequential task execution capabilities greatly increases MDRS functionality.

The accomplishment of a set of sequential tasks requires sufficient information about the progress on the task in order to determine the appropriate action to take at any given time, and in particular at key steps of transitioning between tasks. However, in a MDRS, because of the robots' very limited sensing, intelligence, and communication capabilities, there are many domains in which gathering information on the current state of task progress, part of the global state of the environment, may not be possible for the individuals in the system. Formally, to the individuals in a MDRS, the world is partially-observable and highly non-stationary, yet they must collectively achieve a global goal whose changing state they cannot perceive.

2 Sequential Foraging Task

In the domain of MDRS, the standard, non-sequential foraging task has been studied extensively. We are using a sequential variation of foraging, in order to investigate the capabilities of a MDRS on sequential task execution. Sequential foraging requires a collection of objects (pucks) to be collected in a specified order, based on their color.

Toward proper evaluation of sequential foraging algorithm performance, we developed a cumulative metric that reflects the sequential requirements of the task. The metric, initialized to 0 at the start of every experiment, is updated at every simulation time-step. At each update, for all pucks Puck_{New} deposited in the home region at time t, the utility value, Util(t), is updated according to the procedure:

```
Util(t) = Util(t-1)
for all puck in Puck_New
    if (puck == Puck_Green) then
        Util(t) = Util(t) + Prop_Red
    else if (puck == Puck_Blue) then
        Util(t) = Util(t) + Prop_Red * Prop_Green
```

At the end of an experimental trial, terminated at time t_{Final}, the sequential foraging algorithm is given a final utility value where TP_{Green} and TP_{Blue} are the total number of Puck_{Green} and Puck_{Blue} in the environment, respectively.

$$Util_{Final} = 100 * (Util(t_{Final}))/(TP_{Green} + TP_{Blue}) \quad (1)$$

3 Experimental Environments

The experimental environment consists of an arena with an initial collection of randomly distributed pucks and a home region on one side to which the pucks are to be transported. Whenever a puck is deposited in the home region, it is removed from the arena. We used a group size of four robots and fixed initial locations.

Our experimental design involved the use of four different environment variations on the above arena with characteristics shown in Table 1. The environments were designed to evaluate the adaptability of sequential foraging algorithms along two dimensions: 1) the relative puck type proportions and 2) the arena size.

Env	Arena Size(m)	Red/Green/Blue Pucks
1	8.75x8.75	8/8/8
2	8.75x8.75	14/8/2
3	8.75x8.75	2/8/14
4	17.5x17.5	8/8/8

Table 1: Experimental Environments

4 Sequential Foraging Algorithms

Two foraging algorithms were considered: Timer-Based Foraging and Probabilistic Foraging. These were investigated and analyzed to assess their effectiveness in the sequential foraging task and their adaptability to different environmental characteristics.

In the Timer-Based Foraging algorithm, each robot uses an internal timer to dictate which puck type should be foraged. Each robot has its own independent timer; timers across robots are not explicitly synchronized.

Each robot's timer, Timer_{Robot}, is initialized to 0 at the beginning of an experiment and incremented by 1 at each simulation time-step. A set of timer alarms are used to control which puck types can be foraged at a given Timer_{Robot} value. There is a timer alarm for each puck type: Alarm_{Red}, Alarm_{Green}, Alarm_{Blue}, respectively. When a puck is detected, if the Timer_{Robot} value is greater than the timer alarm for the detected puck type, the robot's Timer_{Robot} value will be reset back to the alarm value of the detected puck type and the robot will begin visually servoing toward the detected puck. A full description of the behavior network used to implement the Timer-Based Foraging algorithm can be found in (Jones and Matarić, 2001).

The Probabilistic Foraging algorithm uses two probabilistic behavior activation conditions in each robot's behavior network. A full description of the behavior network for the Probabilistic Foraging algorithm can be found in (Jones and Matarić, 2001).

The first probabilistic activation condition introduced is whether a robot should visually servo toward a detected puck or ignore the detected puck and perform a random walk. Each robot has an assigned probability of ignoring a detected puck of each type. For the three puck types, these probabilities are: PIgnore_{Red}, PIgnore_{Green}, and PIgnore_{Blue}, respectively.

The second probabilistic activation condition is whether a grasped puck should be dropped before reaching the home region or whether it should continue to be transported toward the home region. Each robot has an assigned probability of dropping a grasped puck of each type while not in the home region. For the three puck types, these probabilities are: PDrop_{Red}, PDrop_{Green}, and PDrop_{Blue}, respectively.

5 Experimental Results

All experiments were performed in simulation using Player (Gerkey et al., 2001) and Stage (Vaughan, 2000) with realistic simulations of the Pioneer 2DX mobile robot.

In the Timer-Based Foraging algorithm, the Alarm_{Red}, Alarm_{Green}, and Alarm_{Blue} values used in all experiments were 0, 750, and 1500, respectively. In the Probabilistic Foraging algorithm, the PIgnore_{Red}, PIgnore_{Green}, PIgnore_{Blue} values used were 0, 0.065, and 0.12. The values used for PDrop_{Red}, PDrop_{Green}, PDrop_{Blue} were 0, 0.65, and 0.12. For each experimental environment and sequential foraging algorithm pair a total of five trials were run. The average Util_{Final} and Standard Deviation values are shown in Table 2.

Alg	Env 1	Env 2	Env 3	Env 4
Timer-Based	99.7/0.5	97.7/0.8	97.0/2.4	79.8/2.3
Probabilistic	96.1/1.6	97.7/2.4	86.7/6.4	98.6/1.4

Table 2: Experimental Results (Util_{Final}/S.D.)

As the experimental results show, the Timer-Based Foraging algorithm adapts well along the dimension varying relative puck type proportions, while the Probabilistic Foraging algorithm adapts well along the dimension of varying arena size. A more detailed analysis of the experimental results can be found in (Jones and Matarić, 2001).

6 Conclusions

The robots in our MDRS maintained little or no state information, extract a limited amount of information from available sensors, and cannot explicitly communicate with other robots in the system. The aim of this work is to provide such a MDRS with the capability of sequential task execution. We presented two experimentally validated sequential task execution algorithms, Timer-Based behavior activation and Probabilistic behavior activation, and experimentally verified and compared them in a sequential foraging task.

References

Gerkey, B., Vaughan, R., Stoey, K., Howard, A., Sukhatme, G., and Matarić, M. (2001). Most valuable player: A robot device server for distributed control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, (IROS-01)*, pages 1226–1231.

Jones, C. and Matarić, M. (2001). Sequential control in a minimalist distributed robotic system. *Institute for Robotics and Intelligent Systems Technical Report IRIS-02-414, Univ. of Southern California.*

Vaughan, R. (2000). Stage: A multiple robot simulator. *Institute for Robotics and Intelligent Systems Technical Report IRIS-00-393, Univ. of Southern California.*