4 LAICHI, F., ABOULNASR, T., and STEENAART, W.: 'Effect of delay on the performance of the leaky LMS adaptive algorithm', *IEEE Trans. Signal Process.*, 1997, **45**, pp. 811–813
5 HAYKIN, S.: 'Adaptive filter theory' (Prentice Hall, 1996), 3rd edn.
6 TOBIAS, O.J., BERMUDEZ, J.C.M., and BERSHAD, N.J.: 'Stochastic analysis of the delayed LMS algorithm for a new model'. Proc. Int. Conf. Acoust., Speech, Signal Process., 2000, pp. 404–407

EDAR – mobile robot for parts moving based on a game-theoretic approach

C.S. Karagöz, H.I. Bozma and D.E. Koditschek

EDAR (event-driven assembler robot) – a mobile robot capable of moving a collection of disk-shaped parts located on a two-dimensional workspace from an arbitrary initial configuration to a desired configuration while avoiding collisions in a purely reactive manner, is presented. Since EDAR uses a higher-level scheduler to switch among the subtasks of moving individual parts, it is viewed as mediating a noncooperative game played among the parts.

Introduction: In this Letter we describe EDAR (event-driven assembler robot) – a mobile robot designed and developed for moving a set of rigid disk-shaped parts from random initial placements to a final assembled configuration. The plethora of robots developed for this task have been based on feedforward approaches using a sequence of (i) plan generation, (ii) trajectory generation, and (iii) control stages [1, 2]. However, the sensor and actuator uncertainties as well as the changing environment have made it imperative that some level of reactivity should be integrated into such strategies [3]. In this Letter, we push this paradigm to the extreme and describe a robot that operates purely on event-driven principles [4]. Doing this represents small progress in developing a complete formalism of reactive systems, which needs to be done if we are to explore trade-offs along the spectrum from feedforward to feedback systems.

Approach: Roughly two ideas are at work in EDAR: artificial potential functions and game-theoretic interpretation of the resulting system. First, the notion of artificial potential functions is employed to encode the subtask of positioning each part to its destination. A standard method of deriving feedback controllers from potential fields is then used to construct a closed loop within each subtask. The advantage of this approach is that it is known that if an artificial potential function has certain mathematical properties [5], then an object moving in a gradient field generated by this artificial potential function will inevitably end at its prespecified goal position without collisions or getting stuck along the way. Since, as has been shown, no single closed loop can result in a completed parts moving task, a higher-level organising principle has to switch between the alternative closed loops. This idea – autonomous scheduling of subtasks – is then addressed using the concept of a game [6]. By interpreting the artificial potential functions determining the closed loop dynamics characterising each of the subtasks as pay-off functions, the higher-level automaton can be seen to be refereeing a game played among the parts to be moved. As there is a set of pay-off functions – one associated with each subtask – the resulting problem leads to a noncooperative game interpretation. As the obstacles presented by the ungrasped parts present a different geometry depending on whether the robot is moving alone or which part it is coupled to, a single artificial potential function no longer suffices to solve the problem. Rather, a class of artificial potential functions needs to be introduced, each encoding one subtask characterising either one robot-to-part motion or one robot-coupled-to-part motion. In this case, the subtasks become conflicting and the higher-level organising principle is interpreted as governing a noncooperative game played among the subtasks:

(i) Next_part: A switching mechanism chooses the next part to be moved by the robot. The robot achieves the subtask of moving this part via a sequence of mate_part and move_part states. Next_part state is re-invoked at the end of mate_part and move_part states.

(ii) Mate_part: A set of feedback controllers, one for each different part, is used. The robot actuation is generated by the negative gradient vector field of the artificial potential function defined for the target part. The robot moves until a minimum point of the function is attained. If the robot stops at a position where its gripper could successfully grasp the part, then a transition occurs to move_part state. However, the robot motion could be blocked on the way before reaching the target part, i.e. a local minimum of this function is attained. In this case, the subtask of moving the part is terminated and a transition occurs back to next_part state to choose another part subtask.

(iii) Move_part: After the robot grasps the part, the coupled object consisting of the robot and the part moves to the goal position of the part. Again, a set of feedback controllers are designed to accomplish this. The robot actuation is generated by the negative gradient vector field of the artificial potential function defined for the mated part. The minimum value of function is attained when the robot moves the mated part to its goal position. However, the motion of the coupled object could be blocked on the way before this goal position is achieved. In both cases, the subtask of moving part is terminated, the robot ungrasps the part and a transition occurs back to next_part state to choose another part subtask.

The switching among the part subtasks are invoked repetitiously in a reactive manner until all the parts are moved to their goal positions by the robot via the following discrete dynamical system: $b[n+1] = f(b[n])$ where b denotes the augmented position vector of the parts and f is the transition map from one blocked part state to the next. This discrete dynamical system can be interpreted as a noncooperative game played by the parts and refereed by the robot. The fixed points of the dynamical system addresses the solutions of the game.

Implementation: EDAR has been designed and constructed with the purpose of implementing this approach (Fig. 1). It is a 2DOF mobile robot with a 3DOF arm mechanism and a 1DOF gripper mechanism. Its projection onto a two-dimensional workspace is a disk. Its gripper is able to hold disk-shaped objects, e.g. pipes, EDAR can sense its joint positions via optical encoders mounted on the joints. The positions and the sizes of the parts are obtained using a stationary camera mounted on the top of the workspace. Its translational velocity is about 6 cm/s and the time required for grasping/ungrasping is about 30 s/part. EDAR moves parts in purely event-driven manner; thus the control software of EDAR is based on the proposed game-theoretic approach.

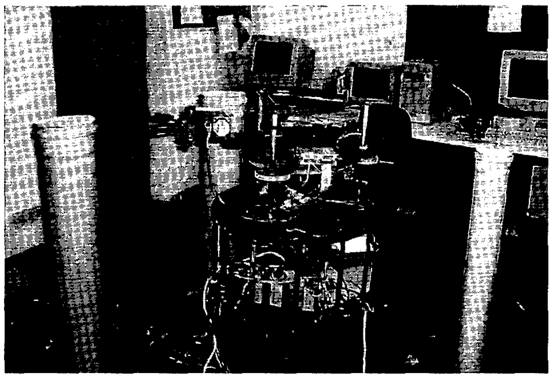

Fig. 1 *Photograph of EDAR with the parts*

Experiments: EDAR has been tested extensively in experiments involving cylindrical pipes of height 1.5 m and of radii varying from 6 to 11 cm. Owing to physical space restrictions, we have been able to conduct experiments involving at most three parts. A typical run of an experiment with three parts is shown in Fig. 2. A set of random goal configurations of varying complexity, measured by the packed tightness of the goal locations, has been defined for the parts. Starting from random initial workspace configurations, the following measures of performance have been studied statistically: (i) the variation of the part path lengths against complexity; (ii) the robot path length against complexity; (iii) the positioning inaccuracy against complexity. Furthermore, experimental results are compared with the simulations of the corresponding tasks.

The following conclusions are obtained. The path lengths of the parts and the robot increase with increasing task complexity. This can be attributed to two factors: first, the closer the parts need to be packed together, the more careful and precise the robot has to be in its movements; secondly, as there is an increased possibility of collision with the parts, there is more dodging around. Interestingly, compared to simulations, the paths taken by the parts are only 10% longer on average (Fig. 3). As expected, path lengths in real experiments tend to vary more owing to the sensor inaccuracies and non-ideal motion capabilities of the robot. The positional inaccuracies in experiments range between 2.0–5.3 cm/part. In simulations we observe much lower inaccuracies which we attribute again to the EDAR sensor and actuator hardware limitations and not to our approach.

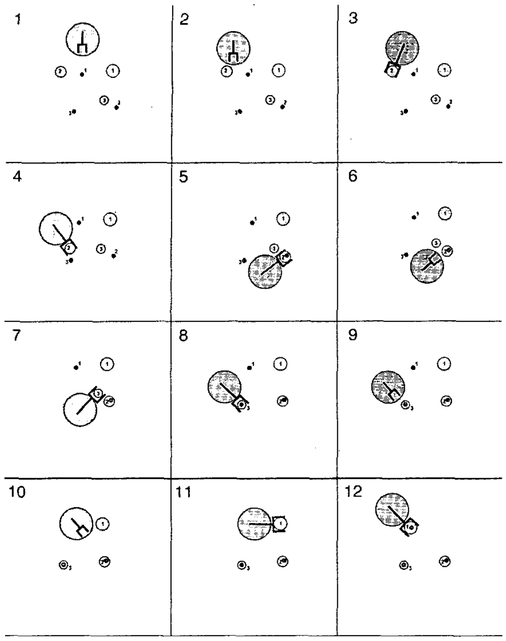

Fig. 2 *Snapshots of experiment with three parts*

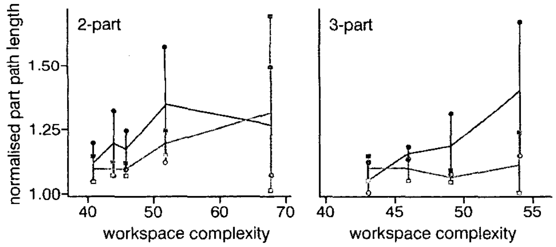

Fig. 3 *Part path length statistics against workspace complexity*

■—□ Simulation data
●—○ Experimental data

These experiments serve to demonstrate that EDAR can successfully complete parts moving tasks, using a purely event-driven strategy.

Acknowledgments: This work is supported by NSF-TÜBİTAK INT9819890 and TÜBİTAK MİSAG65.

14 October 2001
Electronics Letters Online No: 20020089
DOI: 10.1049/el:20020089

C.S. Karagöz and H.I. Bozma (*Department of Electrical and Electronic Engineering, Intelligent Systems Laboratory, Boğazii University, Istanbul, Turkey*)

E-mail: karagose@boun.edu.tr

D.E. Koditschek (*EECS Department, Artificial Intelligence Laboratory, College of Engineering, University of Michigan, Ann Arbor, MI, USA*)

References

1 LATOMBE, J.-C.: 'Robot motion planning' (Kluwer, Boston, MA, USA, 1991)
2 CHA, Y.Y., and GWEON, D.G.: 'Local path planning of a free ranging mobile robot using the directional weighting method', *Mechatronics*, 1996, **6**, (1), pp. 53–80
3 NOREILS, F., and CHATILA, R.G.: 'Plan execution, monitoring and control architecture for mobile robots', *IEEE Trans. Robot. Autom.*, 1995, **11**, (2), pp. 255–266
4 KODITSCHEK, D.E.: 'Task encoding: toward a scientific paradigm for robot planning and control', *Robot. Auton. Syst.*, 1992, **9**, pp. 5–39
5 RIMON, E., and KODITSCHEK, D.E.: 'Exact robot navigation using artificial potential functions', *IEEE Trans. Robot. Autom.*, 1992, **8**, (5), pp. 501–518
6 BOZMA, H.I., and KODITSCHEK, D.E.: 'Assembly as a noncooperative game of its pieces: analysis of 1D sphere analysis', *Robotica*, 2001, **19**, pp. 93–108

Dynamic deficit round-robin scheduling scheme for variable-length packets

K. Yamakoshi, K. Nakai, E. Oki and N. Yamanaka

A dynamic deficit round-robin (DDRR) scheduling scheme for variable-length packets is proposed. It can resolve the drawback of the conventional deficit round-robin (DRR) scheduler that short-packet delay performance and high throughput cannot be satisfied simultaneously. DDRR uses an adaptive granularity for the deficit counter, where the granularity is dynamically changed according to packet lengths in queues. The algorithm, along with the simulation results showing the efficiency, are presented. The DDRR scheduler was implemented for 5 Tbit/s switching system.

Introduction: A fair scheduler for variable-length packets is needed to provide a best-effort Internet-Protocol (IP) service. The deficit round-robin (DRR) [1] scheduling scheme is widely used because of its simplicity to provide the max-min fair share for the best-effort traffic.

However, the DRR scheme cannot satisfy both short-packet delay performance and high throughput, since only a fixed granularity for the deficit counter is allowed. When the granularity is set at too large a value, the short packets are delayed significantly by the long packets. The short-packet delay degradation affects the quality of real-time services such as VoIP. Conversely, the throughput becomes low for a small granularity since the timing margin for the DRR scheduler to satisfy the line speed is reduced as the granularity becomes small.

In this Letter we propose a dynamic deficit round-robin (DDRR) scheduling scheme for variable-length packets. It dynamically changes the granularity for the deficit counter in the scheduler. Simulation results showing the efficiency of the scheduler are presented.

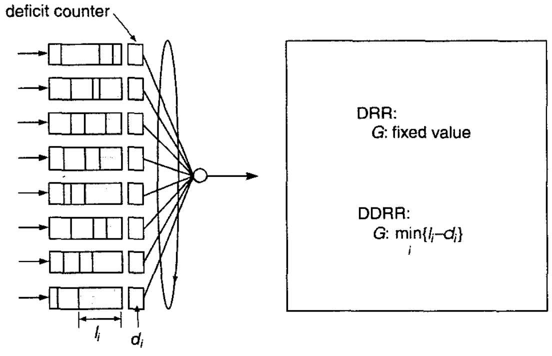

Fig. 1 *DRR-based scheduler for eight queues*

Packet scheduling technique: Fig. 1 is a schematic diagram of a DRR-based scheduler. Fig. 2 shows the average packet-delay dependency on packet length in the case of DRR with the granularity (G) of 5, 10, and 100 cells. Here we assumed that the packet length was normalised to an integer number of fixed-size cell times. The length of