Journal of Intelligent and Robotic Systems **21:** 179–191, 1998.
© 1998 *Kluwer Academic Publishers. Printed in the Netherlands.*

Reactive Robots in the Service of Production Management

ELPIDA S. TZAFESTAS[*]
Intelligent Robotics and Automation Laboratory, Electrical and Computer Engineering Department, National Technical University of Athens, Zographou 15773, Greece;
e-mail: brensham@softlab.ece.ntua.gr URL: http//www.softlab.ece.ntua.gr/~brensham

(Received: 3 April 1997; accepted: 7 July 1997)

Abstract. In this paper, we advocate the use of reactive robots in industrial process control and production management. It is explained why reactive robots are well-suited to modern industrial applications that necessitate a high degree of autonomy and reactivity to unforeseen events and an illustrative example is studied in depth. A reactive robot whose role is to regulate the demands of a set of service units for tool use is described and different behavioral models for it studied and compared. Extensive simulation studies have revealed that a behavioral model relying on individual independent motivations of the robot is sufficient for it to exhibit the "optimal" behavior in this case. The process is modeled as a dynamical system that tends by itself to a steady state and that is perturbed by the robot. In turn, the robot's motivations have a steady state of their own which is contradictory with the process' steady state and hence is perturbed by its presence. As a consequence, the robot and the process are two coupled dynamical systems that perturb each other as each one tries to arrive to its steady state. For the designer, modeling of the overall problem in this way, has as a consequence that the motivational state space of the robot may be designed in advance given the process characteristics and this design should be based on a worst-case analysis. The mythical distinction between reactivity and planning is also rediscussed and the notion of operationality as opposed to optimality is explored.

Key words: behavior-based robotics, reactivity, motivational autonomy, operationality.

1. Introduction

The problems of the design and the design automation of modern manufacturing systems cannot be tackled independently of the underlying system model and architecture. Indeed, often enough, choice of the appropriate model, makes the solution to the design problem transparent. In search of such principles and models that would simplify design, we have been investigating the option of reactive robots and we have been trying to understand to what degree are those robots suitable for modern manufacturing tasks and what additional techniques are required. We have limited our investigation to the process control/production management domain, which among all industrial tasks involves maximal use of

[*] Part of this work has been completed while the author was a Ph.D. candidate in the LAFORIA-IBP (Laboratoire Formes et Intelligence Artificielle, Institut Blaise Pascal, Université Pierre et Marie Curie, Paris, France) under the supervision of Professor Jacques Ferber. Earlier versions of this article appeared as the sixth chapter of [26] and in [25].

non-formal, qualitative knowledge and where various artificial intelligence techniques have been used in the past. The use of reactive robots generally inverses the knowledge representation problem by altogether dismissing the traditional symbolic representation view and relying instead on “know–how” algorithms that themselves “represent” naturally the solution to the given problem. This is possible whenever a holistic interactionist view of the problem/problem-solver system is a better alternative to a piece-by-piece modeling and design. We present a tool-management regulation example where this is precisely the case – a typical specimen of the family of tasks that modern manufacturing is interested in. The possibility of simplified holistic modeling becomes obvious once the robot and the manufacturing process in question are regarded as coupled dynamical systems with conflicting steady states, that constantly perturb each other – a motivationally autonomous robot behaves then as an intelligent controller ensured to function within certain operational bounds. In Section 2, we are reviewing and discussing those aspects of current industrial technology that contribute to the applicability of distributed reactive systems, while in Section 3 we are briefly exposing the reactive behavior-based robots philosophy. Next, we are describing the example process – tool management – and give a comparative analysis of various motivational schemes for the tool manager robot. Some experiments and thoughts on robot specialization are then presented and the paper concludes with a deeper examination of operational principles and requirements behind the feasibility of motivated industrial robotics.

2. The Changing Industry

In the past, robots have been used in industrial contexts to perform tasks that necessitated high precision and accuracy, such as assembly, welding and other mechanical operations. During the last decade, however, research trends in manufacturing systems shifted toward system integration, production flexibility, process adaptivity to unforeseen events, uncertainty handling and fault-tolerance [13, 18, 19, 29]. The concept of cellular manufacturing has emerged as a unifying paradigm that captures the need for product-oriented rather than process-oriented layout of the production plant [19]. Cellular manufacturing relies on allocating the complete manufacturing of a family of parts to a cell, i.e., a group of machines including robots. Such individual cells have to respond to local operational demands and are subject to the constraints of flexibility and adaptivity. *Flexibility* involves the possibility to reinitialize/reconfigure the system on a new set of operational demands, including deadlines to be met, cost limits, order and timing of input etc., while *adaptivity* corresponds to the possibility of recovering from failures of various types. Flexibility therefore concerns mainly the connection of a cell to the other components of the plant, while adaptivity refers to its local self-sustainance. Those constraints have led to a reconsideration of the two fundamental assumptions of conventional manufacturing systems: The first is the *almost-perfect-knowledge assumption*, according to which all or almost all of the parameters that affect the operation and the performance of the system are known in advance – for example, scheduling presupposes that the number, types, durations etc. of jobs are known as well as the number, capacities and possibilities of the processors to which those jobs will be allocated. The second is the *centralized-decision assumption*, according to which – and since there is almost perfect knowledge about the task – the designer of the system can derive a method of controlling it to meet the required operational goals, or else make decisions in advance about the operation of the system. The first assumption has been relaxed over the last decade since no complete information about the overall process is available in advance and the optimization principle has gradually lost ground in favor of more qualitative and heuristic approaches that may manipulate less formal knowledge about the process; those techniques are mainly borrowed from the artificial intelligence domain and their integration into manufacturing domains [2, 14] has paralleled the emergence of the intelligent control field that combined conventional control with artificial intelligence techniques [17]. On the other hand, doubts on the centralized decision have only recently begun to find their way into the scientific arena and some manufacturing systems that involve multiple tasks, robots and resources have started being modeled in terms of concurrently acting cooperative or competitive components, usually called agents (see for example [10]). In such manufacturing systems, decisions have to be made separately at almost every point during operation of the system, since the exact overall state cannot be thoroughly known in advance and long-term planning/scheduling soon becomes obsolete. Local rules or heuristics at various points become more important and complicated analytical pre-processing loses value. Introducing autonomous agents at those decision/redecision points – i.e. points, where some local intelligent, or for that purpose autonomous, "processing" is necessary – looks beneficial. The advantageous consequences of such decomposition of the overall process into a number of interacting agents are *ease of design* (through complexity multiplication during synthesis) and *reconfigurability/scalability*. This multi-agent view of manufacturing systems is in line with Baldwin's [4] view of autonomous manufacturing systems, where goals and intelligence are distributed across all entities (robots, tools, resources etc.) participating in a manufacturing plant.

3. Reactive Robots

During mid-eighties and in an attempt to provide an efficient alternative to classical planning methods (a good collection of papers appears in [1]), the paradigm of reactive systems has emerged. Whereas planning relies on the existence and update of a perfect global world model, reactive systems have adopted the other extreme view of total absence of any internal model. Instead, every action is in fact a direct reaction to a perceived stimulus (stimulus-response or S-R scheme).

However simple in itself this might appear, its power grows exponentially when several reactive components are connected and used within the same system. Interactions between such reflexes may result in what appears to an observer concrete activity or even purposeful reasoning. For the case of a single robot, if S stands for its perceptions and R for the commands finally fed to the actuators, it appears as though something exists in between (S-x-R). This "something" is nothing else but a *Reactive Algorithm* (RA). Reactive because as explained it does not maintain an explicit model of the world nor of the global task (although is *does* have some knowledge about its conditions of operation and about its operation itself) and algorithm because it appears as such and because it was designed so (S-RA-R scheme).

This reactive approach has been found to work best in partially unknown environments, where sensor readings are noisy or unreliable and operation has to be extremely real-time and responsive to unexpected events [6, 7]. Those robots are equipped with a set of competences that they use in order to "survive" in the environment they are put into. Completion of the designers' task occurs as a side-effect of the robots' struggle for survival. In this context, reactive algorithms precisely *minimize processing* of sensor data and accelerate it by exploiting parallelism on fixed topology networks. Another feature of reactive algorithms is that sensor readings or other "stimuli" may arrive at *any point* inside it, what may serve both to monitor command output and to ease design. Other advantages of the use of reactive algorithms include: low implementation cost, design simplicity, enhanced robustness (due to minimal processing and crude data representations), finally potential adaptivity, since a designer may decide to adapt one of the parameters the algorithm uses – there are not too many of them and they are usually visible since the RAs are rather primitive.

Other desirable properties of the reactive, behavior-based paradigm are [6, 7]: no central world representation bottlenecks, minimization of time lags between perception and actuator level command output and fault-tolerance. In one word, the power of the behavior-based paradigm lies in its simplicity and it is left to the interaction with the world to give rise to potentially complex phenomena. On top of these and by exploiting implicit knowledge about the world and the task, we can siginificantly simplify our life as designers and by using such reactive autonomous robots we can build low-cost, robust, fault-tolerant systems that demonstrate conceptual as well as pragmatic ease of design.

Actual use of reactive robots to industrial contexts has already been sketched (see, for instance, [3] and [9]). Such reactive industrial robots, that need to comply with flexibility and adaptivity constraints, have come closer to the recently appeared service robots [20]. In the rest of paper, we will be investigating the use of reactive robots in a production management context and we will demonstrate why the primitive reactive model has to be enhanced with the notions of motivational autonomy and operational coupling.

4. An application: Tool Management and Plant Regulation

We assume an industrial plant that consists of four service units (for instance, assembly cells) that issue demands for tool use. The tool-manager robot goes from unit to unit responding to those demands. Each service unit increments demands periodically up to an upper limit steady state (which corresponds to a unit demanding maximal tool use). Beyond this point, the unit's demand does not increase anymore (the production of this unit is temporarily stopped). Obviously, this arrival process alone tends to saturate the system (all units tend to maximum demand). Our goal is to find suitable reactive robot behavioral specifications that will delay this saturation as long as possible, so as to maximize service units' production (assuming that the robot can't be quick enough to service all units at no time). Demands represent service times. The robot stands initially in its rest place in the center; it repeatedly goes to a unit, services it and returns to the center. Each move from the center to a unit or vice versa needs T_M time units, while each service lasts at most T_S. All the results reported are averages over 10 runs, each lasting 1000 time cycles, with $T_M = 10$ and $T_S = 3$. All simulations have been conducted without and with noise; addition of noise means that with a certain probability – which in the present experiments has been set to 5% – the demand is incremented by 1. This corresponds to failures or other unexpected events in the service unit, that actually delay production and augment tool demand. Most of the experiments used a period of 3 to 5, in which case the default saturation of the system is not too quick and the operational consequences of different motivational control schemes are therefore more visible.

Based on previous experience with reactive robots, we have created one service task for each unit. The tool-manager robot's actuators systems repeatedly evaluate those tasks' motivations and "choose" for execution the one that expresses maximal preference. For this to work, all maximal demands have been normalized to a value d_{max}, which for the present experiments has been set to 40. Each task's motivation is a function of the corresponding unit's demand, such that when demand = 0, motivation = 0. This way, the robot's "nirvana" is when all its motivations are 0; every time its motivations are non-zero the robot tries to act in a way that will bring them back to 0. The presence of the process, that tends to saturate those demands, perturbs the motivational state of the robot and causes it to act/react. The robot control programs are written using a cellular control architecture [24], while the actual high-level pseudo-code run by each service task is:

- If the robot is at the corresponding service unit, service it for a computed amount of time.
- If the robot is at the center, go to service unit.
- If the robot is elsewhere, go to the center.

The tasks' motivations are being evaluated after every such step (move or service). It should be stressed that the navigation and perception systems of the robot have been written once and for all in the beginning and never changed since. Sole intervention in the motivational level of each task has allowed drastic increases in the performance to be observed, as will be shown below.

Note that this is *not* a usual resource allocation problem (see, for example, several papers in [8]): we assume that the individual service units possess a local inventory of tools that they use at will and there is an additional set of tools that need to be shared across units and whose use is sporadic. The tool-manager robot regulates the units' demands for those more "rare" tools, which it is more cost-effective to share rather than possess local instances of them. For this to work, we also assume that individual service units make optimal local use of those tools, for instance by regrouping in time operations that depend on these tools (this flexibility in operation is generally achieved through use of distributed representations of the industrial process/product, such as the assembly Petri nets used in [23] and whose properties have been explored in more detail in [21]). Such a degree of local intelligence in the service units enables us to afford a tool-manager robot that will decide alone when to retreat from a given unit, in which case it will withdraw all tools previously allocated to it, to make them available to other units. Therefore, the effective operation of the units-manager system depends on the existence of such a cooperative coupling. We can certainly proceed to more complex allocation cases, however, by only defining different motivations for different instances of tools, that will be therefore treated and allocated independently. In the following section, we investigate the general principles of use of a motivationally autonomous reactive robot in the simplest case described above.

5. Motivation and Operationality: Post-Reactive Robots

Reactivity has been opposed to the traditional planning approach in an attempt to account for real-time behavior that would not suffer from brittleness of traditional knowledge representation techniques. We believe that time has come for reactivity to be dissociated from the stimulus-reflex concept (originally promoted as the foundation of real-time responsiveness) and to be regarded as purely phenomenal, while in reality it will preserve its real-time character by relying on well-chosen internal "representations" that will ensure operationality. We propose here the animal motivational control theory (a good survey may be found in [22]) as a valid domain of inspiration when it comes to combining reactivity and autonomy with operationality. Related work to this investigation of motivational control schemes includes [5, 16] and [11].

5.1. PERSISTENCE

The first set of experiments were conducted using a naive motivation for each task.

$$\texttt{motivation} = \texttt{demand}$$

This led to the robot never servicing a unit, because at each evaluation of the tasks' motivations, and due to asynchronicity, another one was selected. The solution to this problem is to introduce some persistence: once a task is selected, it should persist a little more beyond the point it has absolute (naive) priority. This mechanism is necessary because the switching costs from task to task are very high relative to the satisfaction gained from each task (here, the T_M cannot be neglected, otherwise the "nirvana" would be immediately attainable).

The first persistence mechanism involved an additive factor to be applied only to the selected task.

$$\mathtt{motivation} = \mathtt{demand} + \mathtt{factor}$$

Experiments with different values for the factor showed that the robot was often stuck to the first unit it serviced, never going to the other ones. Analyzing the above expression for the motivation, we see that this can only happen whenever

$$\mathtt{factor} - \mathtt{T_S} + (\mathtt{T_S}\ \mathtt{div}\ \mathtt{period}) > 0$$

(the second and the third term of the left hand are the quantities of demand serviced and renewed respectively, while a task is active). For period $> T_S$, the above is true whenever factor $> T_S$. Using a factor lower than that only yields a minor improvement, since the robot still abandons a selected task too quickly. A few of the results are summarized in Table I.
Next, a multiplicative factor was introduced, again to be applied only to the selected task.

$$\mathtt{motivation} = \mathtt{demand} * \mathtt{factor}$$

Obviously, that factor should be greater than 1. Experiments with different values for the factor showed that the performance of the system improved substantially – compared to the additive persistence factor – and stabilized with a descending tendency after `factor` = `period`. In the worst case (when all service units are saturated with their maximum demand value d_{max}), it makes sense to have the robot satiate units in turn. In that case, a selected task should persist for as

Table I. Comparative results for the additive persistence case (without noise): number of moves (average service time per unit), average demand per unit. Note that the units are generally saturated, so that even in the case where the robot is stuck (lower right case) the average performance is not much different from the other cases

	Period = 3	Period = 4	Period = 5
Factor = 2	110(57)	109(57.75)	44(23.3)
	36.74	35.336	35.019
Factor = 3	110(57)	65.8(34.95)	1(0.5)
	36.74	35.808	35.844

Table II. Comparative results for the multiplicative persistence case (without noise): number of moves (average service time per unit), average demand per unit. Compare with Table I to see the substantial improvements from the additive case

	Period = 3	Period = 4	Period = 5
Factor = period	37 (156.75) 30.893	35.9 (156.25) 26.498	37 (155.93) 22.127
Factor = 1 + d_{max}	29 (177) 26.975	31 (171.5) 23.205	34 (164) 19.342

Table III. Comparative results for high periods with or without persistence (without noise): number of moves (average service time per unit), average demand per unit. Note that in such high periods the units saturate more slowly than in lower ones

	Period = 25	Period = 30
Without persistence	90(17.825) 11.301	88.8(19.25) 7.415
With persistence	82(37.5) 1.793	84(31.25) 1.507

long as its demand is non-zero, hence $\texttt{factor} > d_{max}$. Comparative results for $\texttt{factor} = \texttt{period}$ and $\texttt{factor} = 1 + d_{max}$ are given in Table II.
Persistence is always imperative, even if the system is not heavily loaded, because it greatly reduces switching costs and hence enhances performance. Results in the case of period = 25 or 30 are evidence to this (see Table III).
To write the corresponding cellular control program, however, we needed a continuous function for the persistence factor, so we adopted the exponential time-convergent

$$\texttt{factor} = 1 + \mathtt{d_{max}}(1 - \mathtt{e}^{-\mathtt{t}/\mathtt{T_p}})$$

which rises from 1 to $\mathtt{d_{max}} + 1$ (t is the time that has elapsed since selection of the task). Since this rise should happen almost immediately after a task is selected, the $\mathtt{T_p}$ constant should be such that $1 + \mathtt{d_{max}}(1 - \mathtt{e}^{-\mathtt{t}/\mathtt{T_p}}) > \mathtt{d_{max}}$, which gives $\mathtt{T_p} < 1/\mathtt{ln}(\mathtt{d_{max}})$. For the present experiments we used a safe value of $\mathtt{T_p}$ approximately equal to $1/2\mathtt{ln}(\mathtt{d_{max}})$, $\mathtt{T_p} = 0.13$ (since $\mathtt{d_{max}} = 40$).

5.2. AUTONOMY

Autonomy means freedom from control. Up to this point, we have introduced persistence mechanisms to make the robot operational in its role. However, the robot is not autonomous yet, since it may be manipulated by vicious/capricious greedy units that will emit maximal demand all the time (this may also be the case of defective units where repeated failures keep the demands saturated). Indeed, simulations have shown that the robot tended to get stuck in such malicious or defective units without ever noticing it was not productive. What appears compulsory is some kind of local "boredom" or decay factor that would make the persistence factor drop gradually back to 1 after a period of elapsed time $\mathtt{T_R}$ (for which, $\mathtt{T_R} > \mathtt{2T_M} + \mathtt{d_{max}}(\mathtt{1} + \mathtt{1/period})$ should hold). In terms of motivational systems theory, we need an additional variable representing an internal incentive: this variable is initialized to $\mathtt{T_R}$ whenever a task is selected and drops gradually with time.

$$\mathtt{factor} = \mathtt{var(t)} * (\mathtt{1} + \mathtt{d_{max}}), \quad \text{with } \mathtt{var(0)} = \mathtt{T_R}.$$

It may be said that this is a hedonistic variable, i.e., one that expresses expected future pleasure; this expected pleasure drops with time, as though the motivated robot was becoming satiated. For permanent failures, an additional decaying factor further toward 0 was introduced, but this is dependent on the types of failures we want to be able to detect and recover from.

6. Specialization?

The same process has been simulated and studied for multiple robots, in search of autocatalytic models of specialization, that would make each of the robots specialize in a fraction of the service units. Autocatalysis means that the motivation of a task depends on a term that is reinforced on execution of the task. Again, a number of experiments have been conducted with additive, multiplicative and multiplicative with an upper bound factors. The second case was far superior to the first, but motivational factors could rise to a very high numeric value; definition of an upper bound solved this problem but resulted in phenomena similar to those observed in the naive persistence case. Combination of an upper bound and persistence as previously defined, has yielded the best results for multiple robots, but still only marginally better than the equivalent case of multiple robots with only persistence. This hardly detectable improvement of the performance with persistent autocatalysis compared with sole persistence can be attributed to the asynchronicity of the system's operation which induces, for example, slightly better regulation of a single-robot 2-unit plant than a 4-unit one that is managed by two robots.

The lesson that may be drawn from the above is that specialization is probably not a good idea in this case and the reason appears to be twofold: i) tasks – that correspond to service units – are *independent*, so that, except for switching costs, no genuine load balancing is necessary, and, ii) mere autocatalysis results to a robot specialized in at most two units/tasks and cannot prove useful in cases such as 3 robots and 8 units. In all cases, persistence alone looks sufficient. If any specialization reveals necessary – not in this case, anyway –, then it will have to rely on more "genetic" causes, rather than on primitive autocatalytic processes, hence on more elaborate sociobiological models [28], rather than on blind reflexes such as autocatalysis.

7. Discussion

The above formulation and study of the adopted process control problem allows a number of interesting observations to be made:

If the control system of the reactive robot is decomposed into a number of activities, or else tasks, that define a desirable steady state, the action selection, or else arbitration, problem may be tackled directly at the actuator systems' level and rely on individual tasks' expressed motivations. The details of separate tasks, such as local adaptation mechanisms, can therefore be addressed independently of the motivational system.

Furthermore, discrete solutions, such as mere presence/absence of motivation are suboptimal. What is necessary is continuous functions at all levels of the reactive control system. Time may be used as an additional, active, indirect control parameter, such as in the case of the time-convergent multiplicative persistence; often enough, the only real measure of success/failure a robot can have about itself is the amount of time spent on a certain activity. As designers of a robotic system, we ought to exploit this. Continuity in the process itself also induces noise to the inputs of the robot's control system. Precisely, all the experiments have been conducted with and without noise and it has been found that the relative performance of the system is a little better when there is noise present than when there is not. Actually, noise is a source of variation and it is known that "*the real cause of stability in a distributed system is sufficient diversity*" [12]. Furthermore, noise in the sense of asynchronicity between process events and robot perception has been found necessary to ensure operationality of the robot's motivational system: for example, we simulated the case of demand increments of 2 with a period of 2 and we found that the robot was always stuck to the first serviced unit, because actually the operation of the system and the robot were synchronous, so that the robot could not *see* (perceive) any worthwhile changes in other units and consequently would not service them.

This study also showed that there should be structural/operational coupling between the robot and the process (the robot's world). This coupling has the sense of using certain process parameters (such as d_{max} and T_M) while designing the robot's motivational control system. The identification of the exact *laws* of this coupling can result from a worst-case analysis of the overall system. Operationality in the above sense means that the behavior/performance of the robot with a given architectural substrate will be "statistically optimal" for the right parameter setting among all available alternatives.

Finally, and since there is process-robot coupling, the robot may be viewed externally as following a purposeful, roughly preset, highly robust and adaptive plan. However, the robot does not manipulate any plans; it merely reacts to the situation at hand. The plan, if any, is in the designer's head (also in the observer's eye): the role of the robot's creator is then to identify the proper operational parameters and devise laws of robot/world coupling that will ensure task completion in such an apparently purposeful way. We have shown in the above simple case that this is feasible: the robot itself doesn't need to plan, because its designer has done it; the idea is that reactivity is the optimal solution in the noisy system case, but it should be based on enough operational knowledge about the overall system (is this evidence for a purposeful world?). In this sense, we are not very far away of the on-line versus off-line debate within the planning community: the designer has taken care of the off-line component which concerns the structure of the motivational control system, while interaction with the world will provide the actual data that will be used by the on-line decisional process. Completely reactive robots have to be reconsidered in this case as *intuitive robots*, robots that are preprogrammed with the right reactive "knowledge" potential; this potential is what Verschure [27] calls a value system. One trick of that mechanism is the presence of evaluation points in the motivational system: as is described before, the motivations of the robots are re-evaluated only at certain points and not continuously. On the one hand, this is to account for local noise that might make the robot oscillate between activities, while on the other hand it concerns an implicit system assumption: the robot can afford checking its motivations with considerable lags because the individual activities are adaptive and robust by themselves.

8. Conclusions and Perspectives

We have briefly exposed observations and experiments that lend support to the proposition that reactive, motivationally autonomous robots are a conceptually and operationally valid candidate for design of modern manufacturing systems. The approach relies on the modeling of the robot(s) and the controlled process as two coupled dynamical systems with conflicting steady states that constantly perturb each other. The fortunate consequences of such a modeling is that the motivational state space of the robot may be designed in advance given the process characteristics and this design should be based on a worst-case analysis. The above also suggest that reactivity and planning are not as mutually exclusive as is generally thought: the question shouldn't be, does it react or does it plan, but rather, who plans for it? In our case, we have shown that if we, designers, understand sufficiently the whole process, we can build into a robot a completely reactive behavioral code that will ensure task completion. This may be viewed from the outside observer as a reactive, robust plan undertaken by the robot, but it is nothing more than the right reaction pre-programmed into it by its designer. The notion of optimality has been also reconsidered in this case. Ongoing work includes a quantitative study of the dynamics of the system, in the same line as that followed in [15].

Acknowledgement

I thank Philippe Darche for continuing discussions and robot implementation tips.

References

1. Allen, J. F., Hendler, J., and Tate, A. (eds): *Readings in Planning*, Morgan Kaufmann, San Mateo, CA, 1990.
2. Almgren, R.: On Knowledge-Based Planning and Programming Systems for Flexible Automatic Assembly, University of Linköping, Studies in Science and Technology, Thesis no. 176, LiU-Tek-Lic-1989:16.
3. Arkin, R. C. and Murphy, R. R.: Autonomous navigation in a manufacturing environment, *IEEE Transactions on Robotics and Automation* **RA-6**(4) (1990), 445–454.
4. Baldwin, K. E.: Autonomous manufacturing systems, in: *Proceedings of the 1989 IEEE International Symposium on Intelligent Control*, Albany, NY, September 1989, pp. 214–220.
5. Balkenius, C.: Motivation and attention in an autonomous agent, in: *Presentation Workshop on Architectures Underlying Motivation and Emotion – WAUME 93*, University of Birmingham, 1993.
6. Brooks, R. A.: Intelligence without representation, *Artificial Intelligence, Special Issue on the Foundations of Artificial Intelligence* **47**(1–3) (1991), 139–159.
7. Brooks, R. A.: Intelligence without reason, M.I.T. AI Memo No. 1293, April 1991, also *Computers and Thought Workshop, IJCAI-91*.
8. Desrochers, A. A. and Silva, M. (eds): *IEEE Transactions on Robotics and Automation, Special Issue on Computer Integrated Manufacturing* **10**(2) (1994).
9. Doty, K. L. and Van Aken, R. E.: Swarm robot materials handling paradigm for a manufacturing workcell, in: *Proceedings of the 1993 IEEE International Conference on Robotics and Automation*, Atlanta, GA, May 1993, pp. 778–782.
10. Fischer, K.: Modelling autonomous systems in a flexible manufacturing system, in: *Proceedings of the IJCAI '93 Workshop on Dynamically Interacting Robots*, Chambéry, France, August 1993, pp. 109–116.
11. Halperin, J. P. R.: Cognition and emotion in animals and machines, in: H. L. Roitblat and J.-A. Meyer (eds), *Comparative Approaches to Cognitive Science*, MIT Press, 1995, pp. 465–500.
12. Hogg, T. and Huberman, B. A.: Better than the best: The power of cooperation, in: L. Nadel and D. Stein (eds), *SFI 1992 Lecture Notes in Complex Systems*, Addison-Wesley, 1993, pp. 163–184.
13. Jones, P. F.: *CAD/CAM: Features, Applications and Management*, Macmillan, London, 1992.
14. Kusiak, A.: Manufacturing systems: A knowledge- and optimization-based approach, *Journal of Intelligent and Robotic Systems* **3** (1990), 27–50.
15. Levi, P. and Will, T.: A study of the dynamical behaviour of a self-organised production system, *Poster Presentation, Artificial Life IV Workshop*, Cambridge, MA, July 1994, 6 p.
16. McFarland, D. and Bösser, T.: *Intelligent Behavior in Animals and Robots*, MIT Press, Cambridge, MA, 1994.
17. Meystel, A.: Intelligent control: A sketch of the theory, *Journal of Intelligent and Robotic Systems* **2** (1989), 97–107.
18. Ranky, P. G.: *Computer Integrated Manufacturing – An Introduction with Case Studies*, Prentice-Hall, Englewood Cliffs, NJ, 1986.
19. Rotstadås, A. (ed): *Computer-Aided Production Management*, Springer, Berlin/Heidelberg, 1988.
20. Schraft, R. D., Degenhart, E., and Hägele, M.: Service robots: The appropriate level of automation and the role of operators in the task execution, in: *Proceedings of the 1993 IEEE Systems, Man and Cybernetics Conference*, pp. 163–169.
21. Suzuki, T., Kanehara, T., Inaba, A., and Okuma, S.: On algebraic and graph structural properties of assembly Petri nets, in: *Proceedings of the 1993 IEEE International Conference on Robotics and Automation*, Atlanta, GA, May 1993, pp. 507–514.

22. Toates, T.: *Motivational Systems*, Cambridge University Press, Cambridge, UK, 1986.
23. Tzafestas, S. G. and Tzafestas, E. S.: The blackboard architecture in knowledge-based robotic systems, in: T. Jordanides and B. Torby (eds), *Expert Systems and Robotics, NATO ASI Series Vol. F71*, Springer, Berlin/Heidelberg, 1991, pp. 285–317.
24. Tzafestas, E. S.: A cellular control architecture for autonomous robots, in: *Proceedings of the 1994 International Workshop on Intelligent Robotic Systems*, Grenoble, July 1994, pp. 70–79.
25. Tzafestas, E. S.: Reactive robots in the service of production management, in: *Proceedings of the 4th International Conference on Intelligent Autonomous Systems (IAS-4)*, Karlsruhe, March 1995, pp. 449–456.
26. Tzafestas, E. S.: Vers une systémique des agents autonomes: Des cellules, des motivations et des perturbations, Ph.D. thesis, Université Pierre et Marie Curie, Paris, December 1995.
27. Verschure, P. F. M. J.: Formal minds and biological brains, in: *IEEE Expert*, October 1993, pp. 66–75.
28. Wilson, E. O.: *Sociobiology – The New Synthesis*, Belknap Press/Harvard University Press, Cambridge, MA, 1975.
29. Wright, P. J. and Bourne, D. A. (eds): *Manufacturing Intelligence*, Addison-Wesley, Reading, MA, 1988.