What is Networked Robotics?

Gerard McKee

Active Robotics Laboratory, School of Systems Engineering, The University of Reading, Reading, Berkshire, UK, RG6 6AY
g.t.mckee@reading.ac.uk

Abstract. Networked Robotics is an area that straddles robotics and network technology. A robot system controlled via the WWW exploits the Internet network and hence is one realisation of networked robotics. A set of field robots that exploit wireless networks to share and distribute tasks might also be considered an exemplar of networked robotics. But isn't this just an exemplar of distributed robotics? And if so, what does networked robotics bring to the "robotics" table? These are questions and issues addressed in this chapter. The chapter will propose that networks are at once both enabling and constraining to robotics. They enlarge the scope of the robotics discipline yet introduce challenges that must be overcome if that potential is to be fully realized. In short, when the network becomes a design issue – normally when performance of the system is at a premium – networked robotics is at play.

Keywords. Networked robotics, distributed robotics, robot architectures.

1 Introduction

Computer networks are a pervasive element of everyday life. They allow us to access information at a distance and indeed to link geographically dispersed computing resources into powerful, distributed computing platforms. The World Wide Web (WWW) motivated the first integration of robotics with networking through the creation of online robot systems [12]. Teleoperation, which has a well established record of operating robots remotely, subsequently found a new transport medium in which to explore remote control issues and novel applications [25] [13]. Client-server models, object orientation and multi-agent systems offered robotics the opportunity to compose complex systems from distributed sensor, effector and computational resources [21] [8] [9]. More expansive views envision networked robotics as humans and robots acting together in intelligence spaces [15].

The online robot system, the Internet robot, the distributed robot architecture and the intelligence space are all very different views of networked robotics. What, however, are the underlying principles and purposes of networked robotics? What exactly is networked robotics? What or how does it contribute to robotics and, indeed, to networking, if anything? Or is networked robotics just the latest in a long line of fund-worthy buzzword? These are the questions that this chapter sets out to address. The important message is that networked robotics is an enabler – it allows us to do things that were not previously possible with robots – the online robot is just one example. However, it is also constraining, since in order to do many of these things there are problems that we need to overcome. Ultimately it is a liberator, for in doing these things and overcoming these problems, robot architectures and systems can be liberated from the constraints of fixed wiring.

The remainder of this chapter is organized as follows. The following section presents the component or modular view of robotic systems that is emphasised in the treatment of networked robotics provided in this chapter. Section 3 then makes the distinction between robotics resources, robot systems, robotic agents and tasks that follows from this modular view. Section 4 compares networked robotics with distributed robot systems, introducing robot talk networks and distributed robot architectures. Section 5 widens the discussion to include ambient and pervasive intelligence, and in this context as well sensor and other forms of distributed networks of devices and computational nodes. Section 6 incorporates internet and online robots into the framework developed in the previous sections. Section 7 explores networked robotics within the context of field robotics. Section 8 provides a general framework for exploring important issues in networked robotics. Section 9 concludes the chapter. The chapter does not offer a review of networked robotics but only dips into research that has a bearing on the question posed in the title.

2 Breaking the Robot Apart

Early robot systems incorporated deliberative reasoning architectures in which the functional elements of the architecture, such as vision analysis and path planning, followed sequential programming principles and interacted through procedure calls [22]. The systems were by today's standards monolithic, undifferentiated and unstructured. Indeed, they rapidly lost their appeal when more fine-grained behaviour and hardware-oriented approaches to building robot systems emerged [3]. While the deliberative and the behaviour-based approaches both offered opportunities for modularity, and the latter was heralded as a more modular and hence more flexible approach to systems engineering, they were both slow to follow when the wider discipline of computing embraced object-orientation [7] – the additional effort implied by object orientation was viewed as unnecessary and would only get in the way of engineering the robot system. However, object orientation did appeal to some researchers [10] [23] and the importance of modularity was not lost on the designers of reconfigurable robot systems [1].

Networked robotics is founded on the ability, at different levels of granularity, to model a robot system as a set of components that are glued together to compose robot systems. Object orientation is useful to networked robotics to the extent that it fosters the idea of a software component that encapsulates a certain unit of functionality and provides a well-defined interface by which other objects can avail themselves of that functionality – by way of a set of services on the interface. Therein lies part of the distinctive character of networked robotics and the challenges it faces: What defines a useful component? How are robot architectures composed from these components? Robot systems lend themselves to component-based approaches when we enumerate their sensors and effectors, but it is less obvious how to break apart the rest of the system. This is a key challenge for robotics engineering in general but is specifically addressed by networked robotics in the context of distributed architectures.

Once we tender the notion that a robot system can be modelled as a set of components that interact with each other we can then ask the question as to how these components are represented on a computer system; as the elements of a suite of libraries that are glued together under an application that is compiled and executed; or as active components whose interfaces are made public through a registry service or its equivalent. The latter model fosters the further idea of automated reconfiguration of the system. Reconfiguration is a familiar idea to robotics, but primarily in the guise of modular reconfigurable robot systems [14]. Networked robotics makes the model more accessible to the wider robotics community.

Why would one wish to remap components to create alternative robot architectures? The simple answer is that it is one of the possibilities that an active component-based approach to robot architectures offers. It is not a recognised path to follow in robotics since the standard model for robot systems engineering is to identify a task and to create a robot system to support that and only that task. The task may involve a number of distinct stages that require swapping in or out desired our undesired functionality [4] [3]. In general, however, the one architecture is expected to persist for the duration of the task and indeed indefinitely. The argument is typically offered that this persistence necessarily concedes optimization for specific subtasks in favour of optimization across the whole task. However, reconfiguration may offer a better approach, a least an alternative, and networked robotics makes reconfiguration accessible.

3 Tasks, Agents and Resources

In robotics we can talk of a robot system, typically a mobile robot platform, embodying a set of robotics resources. The resources include sensors, effectors, and computational components. Robot architectures bind together these components in a form suitable to complete some designated task. Under networked robotics the relation between the task and the resources is the architecture. The architecture coordinates the resources to complete the task. It is the embodiment of robotic agency. The role of the robot platform in this model is only incidental: it mounts the physical sensors and actuators and typically as well the computational platform on which the computational components execute. Therefore, in networked robotics we associate tasks with agents and in turn associate agents with resources, not the robot platform. This gives networked robotics it's wider reach to sensor networks and ambient intelligence.

The networking in networked robotics thus eliminates the requirement for the resources that are coordinated to be localised to a single robot platform. The components are enhanced through the awareness of the network and the ability to perform remote procedure calls or adopt process-oriented message-passing models for interaction with other components. In short, the components are network-enabled. The robotic engineer is no longer constrained to map robotic agency one for one to a mobile robot platform, but that option is not conceded either – the engineer now has more options.

4 Robot Talk Networks

We will now draw a distinction between networked robotics (NR) and distributed robotics (DR). Research on multirobot systems covers a very broad area of robotics [2], but this chapter will focus on the specific notion of a team of mobile robots cooperating to complete some task [24]. The distributed in DR then implies multiple robot systems geographically dispersed about the task environment. A network framework is often used in these systems to allow the robots to communicate and indeed to allow one or more human operators to guide the system as a whole or the robot individually. The key difference between NR and DR is that in the former the robot platform is modelled as a physical cluster of robotic resources whereas in the latter the robot platform is a modelled as a single robotic agent.

Given the above interpretation of DR, the problem statement for DR is then: *Given a set of robots and a task that has been broken down into a set of subtasks, how does one allocate these subtasks between the set of robots in order to achieve optimum performance?* NR, on the other hand, takes robotics resources as its primitive architectural unit, not the robot platform, and hence the problem statement is formulated accordingly: *Given a set of robotics resources and a task which can be broken out into subtasks, how does one compose the resources to create robotic agents that will perform the task?*

When the problem statements for NR and DR are defined in this way there is, of course, no explicit reference to networking. The network is really just a part of the implementation. In fact, the NR problem statement is really a statement about distributed robot architectures (DRA). By this we mean that the components of the architecture are distributed across multiple robot platforms and the network provides a means of wiring the components together. This can be contrasted with the role of networking in DR, where it effectively provides a “talk network” which allows robots to talk with each other as and when the need arises.

The basic unit for communication within a robot talk network is a robotic agent. The agent-to-agent communication within a task setting will typically be high-level, for example passing back and forth higher-level observations, negotiating for resources or cooperating on task planning. The basic unit of communication for distributed robot architectures is a resource-to-resource network data connection, which can be referred to as an *arclet*, representing a wired connection. Since the latter is part of the control architecture of the robotic agent, and perhaps requires coordinated action across multiple robot platforms, there may be non-functional performance requirements that need to be satisfied. These requirements may in turn place considerable pressure on the performance of the network, and hence cause the engineer to consider network performance in the design of the networked robotic agent. These requirements are much harder to deal with than the requirements for talk networks, which may simply rest on maintaining agent-agent network connectivity.

5 Embedded Intelligence

Networked robotics embodies the requirement to make explicit the physical, geometrical and functional properties of sensors and effectors in order to allow distributed sensors and effectors to be integrated in networked robotic agents. In doing so it affords as well the opportunity for robotics to exercise a form of outreach into new application areas.

There is no reason, for example, why the resources that are integrated should not include sensors and effector devices embedded in the environment in which a mobile robot operates, or indeed in which humans live and work. In fact, there is no reason why the robotic intelligence should not be based exclusively on sensor and effectors systems embedded within the environment. The contrast can be drawn, therefore, between embodied intelligence (i.e. intelligence embodied in robotic artifacts) and embedded intelligence (i.e. intelligence embedded in the environment).

Of course, what is environment and what is object are relative – a set of mobile robot systems may provide an embedded robotic intelligence to one or more mobile robot systems or humans that they collectively enclose. The latter concept allows us to draw the link, in turn, between network robotics and sensor networks, and ultimately to other forms of device networks through to an all-encompassing ambient and pervasive intelligence. Some of the most innovative work in networked robotics addressed precisely this form of intelligence [15]. These links are important in opening up new opportunities for robotics beyond the traditional industrial robot systems.

6 Internet and Online Robots

The model of networked robotics outlined above is largely one of resources and distributed architectures. Where in this lies Internet and online robots? Online robots have their origin in web-based access to remote laboratory-based robot systems [12], [20]. Internet robotics, on the other hand, has its origins in teleoperation, telemanipulation and telerobotics [25]. It has simply transposed the medium of communication to a setting, the Internet, which is outside the designer's control [13]. In the case of Internet robotics time delay is a major issue, whereas in the case of online robots it is generally not such a major concern. Failure to gain control of time delays in the first instance can result in the failure of the system due to instability, whereas in the second case it can lead to annoyance but generally not complete failure. If the Internet robotic system is part of a remote surgical system, for example, then failure may have severe consequences. Failure of an online robot experiment can lead to frustration, waiting around, but generally nothing catastrophic.

Since the performance of the Internet is a factor that needs to be taken into account in the design of online robot systems and internet robots, though for different reasons, these systems are central to networked robotics. The key challenge, for online robots as well but to a lesser extent than Internet robots, is that fluctuating bandwidth, time delays and network jitters are givens [17]. The obvious approaches to addressing these problems is to reduce the time dependency of the system by incorporating more local intelligence at the robot site (telerobotics versus teleoperation), designing new communication protocols that adapt better to changing bandwidth limitations and requirements, or adapting both the operational speed and performance of the operator interface and remote robot system in tandem with changing bandwidth availability and requirements.

There are two major challenges for online robot systems. The first is to build interesting, reliable, online robot systems that engage the public and students alike, and which can operate 24/7. For use in robotics education they must also incorporate assessment methods within a learning programme. The latter, however, requires the development of a robotics curriculum if the system is to make a substantive contribution to robotics education [18].

The second major challenge is in fact one that requires both Internet and Online Robots techniques. This is precisely the concept of open research laboratories; robot laboratories, that is, that offer a particular research and development challenge, requiring remote contributors to have access to laboratory systems under manual (individual or shared) or mediated control, that provides facilities to allow users to contribute software and hardware components that can be combined to build robot architectures, and offers learning resources introducing laboratory systems and indeed the discipline of robotics. Allied to this is the creation of protocols by which components interact.

7 Field Robot Systems

The Internet is a highway that anyone and everyone can use. There is little scope for robotics to exert an influence on its design in order to better serve the requirements of Internet and online robots. In field robot systems, where a set of robot systems are typically deployed in a natural environment, the possibilities are very different. Such systems, even if networked via wireless Ethernet, can be isolated from the Internet highway. In these setting the range of networking options can be expanded to include radio and even infra-red networks, and the network can be put under tighter control. Indeed, field robot systems are better served by a more networked robotics centred viewpoint that recognises the importance of distributing resources in sensor and actuator networks [6] [16].

Heterogeneous network environments offer new possibilities for networked robotics. For example, radio networks, since they tend to have low protocol overheads can support multi-channel real-time communication requirements between low-level control systems on geographically dispersed robot platforms. Ethernet, on the other hand, can support a form of robot talk network allowing the robot systems to share information and knowledge, distribute tasks and to effect such operations as role-transferral. In order to exploit this opportunity, and to explore distributed robot architectures, as against just robot networks, it is important to gain control not only of the network but also of the processor nodes that support the computational elements of the robot architecture.

Field robotics presents other interesting opportunities and challenges for networked robotics. For example, a human operator may need to interact with the set of robots either individually or as a unit and to a greater or lesser degree. A networked robotics agent may be embedded within the field robot system to mediate between the human operator and the robots. The agent may be distributed across the set of individual robot systems; giving it a presence in every robot system; but nevertheless it has an existence independent of the individual robots.

In a conventional robot team setting the mediating intelligence is typically run on a separate server and instructions or state changes are communicated via a wireless network to the individual robots. Robot soccer teams that participate in the RoboCup competition are one example [11], in fact these represent a case of field robotics in the small. The network robotic agent may in fact not only monitor the state of the robot system but also of the environment and change the state of the robot team accordingly (e.g. from defence mode to attack mode).

These, and other, possibilities are a consequence of fully exploiting networking within robotics. They are afforded by the introduction of networking, but, as stated, they also raise problems that need to be overcome if those possibilities are to be realized. The networking problems for field robot systems in the large, in addition, are very different to those for field robot systems in the small. Indeed, although one might consider that such possibilities are afforded by network technology, the question is whether indeed they really are possible and then as to whether they are useful is a challenge to both networked robotics and to robotics engineering.

8 A Platform Model for Networked Robotics

Networked robotics is concerned with engineering the underlying platform that supports the implementation of robot systems, whether networked, distributed, field, reconfigurable, internet, or online robot system. This section will offer a platform model based on network robotic concepts introduced in previous sections. Figure 1 gives a diagrammatic view of the model.

The model is centred on a modular approach to robotics, whereby a robot system is modelled as a hierarchy of modules. These comprise primitive modules and glue modules. Primitive modules are resource modules – they offer well-defined services. Glue modules (solid circles in Fig. 1) incorporate abstract task requirements and link with resource modules to realise these requirements. They can not only offer the functionality embodied in the module they link to, but also create new functionality. When linking has been completed the glue modules become resource modules. The hierarchy, so crafted, can be constructed to the appropriate level required for the services of an Internet or online robot, or still higher to networked and distributed robotic systems.

The composition of robot architectures can follow traditional manual methods based on compiling and linking code, and process-oriented or remote procedure models for concurrent, decentralised, or distributed applications. Networked robotics, however, offers an alternative model for the composition process. In this model the modules are network enabled, and both self and context aware. Network enablement means that the modules have a unique network address and can listen to and connect with other modules across the network. Self-awareness means that the modules are aware of their own capability, specifically the services they offer and under what

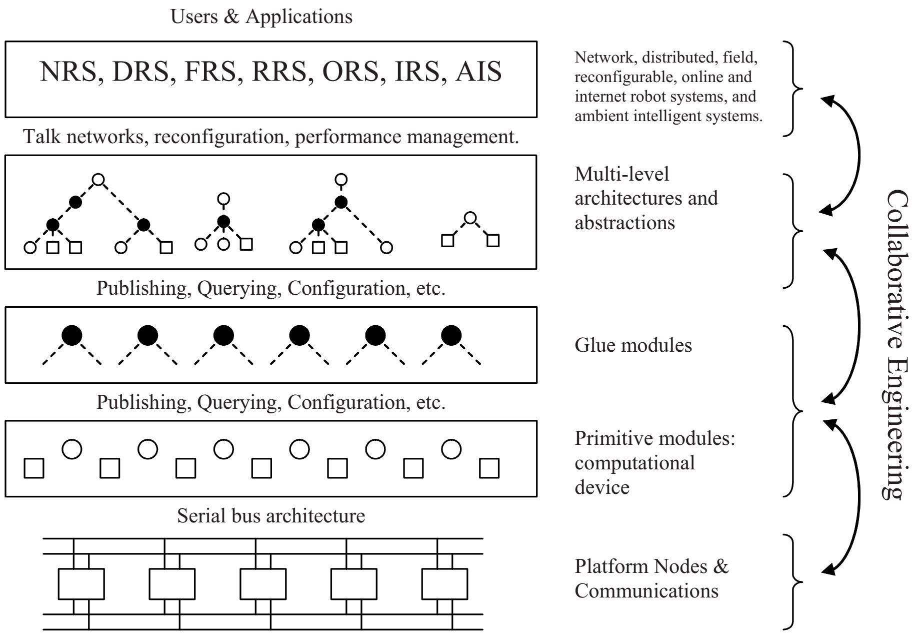

Fig. 1 A platform model for networked robotics.

conditions they can offer these services, and the underlying platform resources (processing and communication) required to provide these services. Context awareness means that the modules are aware of their network and physical location. The latter include, for device modules for example, their physical location on a mobile robot platform.

Primitive modules by definition are assigned a location on the network. So too are glue modules, prior to and following instantiation. In effect, every module is exposed to the network to form a flat serial bus architecture from which application hierarchies can be composed. Additional platform level services are required to support this composition process, including the ability for modules to publish or advertise themselves, for glue modules to query the set of published modules for those that can help them to satisfy their functional and non-functional requirements, and for negotiation to gain access to module services. These services must also be reflected in the design of the modules [5].

At the highest level the architecture forms single or multiple robotic agents in the form of networked, distributed, field, reconfigurable, online, and Internet robot systems (NRS, DRS, FRS, RRS, ORS, and IRS respectively in Fig. 1). These in turn offer application services to users. The modules that form the architecture live within a platform environment comprising process nodes and communication pathways. These platform-level resources need to be properly deployed in the service of the application. This includes the ability for modules to move about in order to satisfy performance requirements. Here one finds typically two opposing forces acting: the first, dispersal of modules in order to balance the load across the processor nodes; the second, clustering of modules on the same processor node in order to satisfy real-time requirements. As modules come to life, move, or die, the available processor and bandwidth resources will change. The modules need to be aware of these changes and react appropriately: knowing when to stay put, to ask for more resources, or to move to other nodes. In this the modules are situated and need situated intelligence to get the best out of the platform for the application.

Platform services must be provided to support the mobility of the modules. These services must include as well a model of the capacity of each processor node and of the communication pathway so that the pressure on resources can be assessed. Reconfiguration of the platform nodes and of the communication pathways as well will provide more scope for optimizing the platform resources available for the application systems. Another set of reconfiguration services are also required to allow the reconfiguration of networked robotic robot systems. Such reconfiguration may be required to meet the requirements of different subtasks. The graceful transitions between these different systems will require additional services.

The modular, networked robotics approach also offers benefits for the process of developing robot systems and applications. The modular, networked approach means that engineers can develop and publish individual modules, create subsystems and offer both of these to others, giving a more open, collaborative engineering environment. This is a natural extension of the online robot and the Internet robot systems concept into an important and challenging area of engineering [19].

Networked Robotics in the model above is first and foremost a medium for wiring together robotics components to form robot architectures. The operation and performance of the platform are important in maintaining the appropriate performance and stability of the architecture. Internet and online robots are special, important, and popular representations of such architectures. However, they are only the surface. The network also offers avenues for integrating robotics technology with other newer network based technologies to create ambient and pervasive robotic intelligence. The platform model above reflects this, but extends the scope beyond hardware-based distributed devices to systems comprising both hardware and software components, and systems offering opportunities for reconfiguration.

9 Conclusions

Network technology is both enabling and constraining for robotics. It offers new opportunities, such as online robots, but it creates challenges that need to be accounted for either in the robot design or in the creation of new network environments. The extent to which the network is taken into account in the design of the robot system determines the extent to which the work contributes to networked robotics. Connecting a set of robots or robot components to a network is only incidentally networked robotics. Incorporating measures to accommodate for network time delays in the controller of a telemanipulation system is hard networked robotics. Gaining control of the underlying network and processor platform, creating new protocols and heterogeneous networking options, is currently only realistic within field robot systems, but is core networked robotics.

Networked robotics allows us to take further the progress that has led from the early structured yet monolithic robot control systems through the object-oriented software paradigm that meshes well with robots at the device level, to active component and ultimately self-aware network enabled modules that can broker their own connectivity. The glue logic that enables primitive robotic resources to be blended into successively higher-level functionality needs to be modelled. A framework must be provided that allows the components to be mapped and remapped between robot architectures. The creation of such a framework is an important challenge for networked robotics.

Networked robotics is little concerned with the content of the components that the robotic engineer defines. It is concerned with content and hence robotic intelligence to the extent that this content needs to be communicated to other modules. This is part of the self-awareness that needs to be built into modules. Networked robotics is only incidentally distributed robotics to the extent that the latter incorporates a network as a medium for the robots to talk to each other. However, a networked robotics environment may be configured as a distributed robotics systems. In this is it is clearly a more general class of system. Networked robotics is more than incidentally cooperative robotics to the extent that a cooperative robotics system incorporates distributed robot architectures and communication resources are explicitly modelled.

Online robots are a popular example of networked robotics, but networked robotics is much more. The purpose of this chapter was to explain how much more. Networked robotics is about modularity, it is about using network technology to wire together distributed, networked enabled, self and context-aware modules that offer well defined sets of services. Networked robotics is about composing robot architectures that express themselves through robotic intelligence in robot-robot and human-robot systems and interactions. Networked robotics offers robotics a timely injection of insight and innovation; we have the opportunity to reflect on the discipline of robotic science and engineering and at the same time engage new technologies as users and enablers. Networked Robotics is an opportunity. Once we have understood better what robotic science is about and what robotics can do, and in particular once we have a new set of tools for composing robot systems, we can set networked robotics aside.

References

1. Anderson, R. J., 1995. SMART: A modular control architecture for telerobotics. In *IEEE Robotics and Automation Magazine*, September, pp. 10–18.
2. Arai, T., Pagello, E., and Parker, L. E., 2002. Guest Editorial, Advances in Multirobot Systems. In *IEEE Transactions on Robotics and Automation*, Vol. 18, No. 5, pp. 655–661.
3. Arkin, R., 1998. *Behaviour-Based Robotics*, MIT Press.
4. Arkin, R. C. and Balch, T., 1997. AuRa: Principles and Practice in Review. In *Journal of Experimental and Theoretical Artificial Intelligence*, Vol. 9, pp. 175–189.
5. Baker, D., 2006. DINERO: Distributed networked robotics. PhD Thesis, The University of Reading, UK.
6. Batalin, M. A. and Sukhatme, G. S., 2003. Using a Sensor Network for Distributed Multi-Robot Task Allocation. In *IEEE International Conference on Robotics and Automation*, New Orleans, LA, April 26–May 1, pp. 158–164.
7. Booch, G., 1993. *Object Oriented Analysis and Design with Applications*, Benjamin Cummings.

8. Brooks, B. G. and McKee, G. T., 1997. Agent-based control architecture for teleoperation. In *SPIE Proceedings Volume 3209, Sensor Fusion and Decentralised Control in Autonomous Systems*, pp. 200–211.
9. Brooks, B. G., McKee, G. T., and Schenker, P. S., 2001. The Visual Acts Model for Automated Camera Placement During Teleoperation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Korea, pp. 1019–1024.
10. Chochon, H., 1991. Object-oriented design of mobile robot control systems. In *2nd ISER*, Toulouse, France, pp. 317–328.
11. D'Andrea, R., 2005. The Cornell RoboCup Robot Soccer Team: 1999–2003, in D. Hristu-Varsekelis and W. S Levine (eds), *Handbook of Networked and Embedded Control Systems*. Birkhauser, pp. 793–804.
12. Goldberg, K., 2000. *The Robot in the Garden*, Cambridge, MA, MIT Press.
13. Goldberg, K. and Seigwart, R., 2002. *Beyond Webcams: An Introduction to Online Robots*, The MIT Press.
14. Kim, M., Roufas, K., Duff, D., Zhang, Y., Eldershaw C., and Homans, S. B., 2003. Modular Reconfigurable Robots in Space Applications. In *Autonomous Robots* Vol. 14, (Nos. 2–3), pp. 235–237.
15. Lee, J-H. and Hashimoto, H., 2003. Controlling mobile robots in distributed intelligent sensor network. In *IEEE Transactions on Industrial Electronics*, Vol. 50, No. 5, October, pp. 890–902.
16. Makerenko, A., Nettleton, E., Grocholsky, B., Sukkarieh, S., and Durrant-Whyte, H., 2003. Building a decentralized active sensor network. In *Proceedings of ICAR*, pp. 332–337.
17. Liu, P., Meng, M., and Yang, S., 2003. Data Communications for Internet Robotics. In *Autonomous Robots*, Vol. 15, No. 3, pp. 213–223.
18. McKee, G. T., 2006. The maturing discipline of robotics. In Special Issue of the *International Journal of Engineering Education* (IJEE) on Robotics Education, www.ijee.dit.ie.
19. McKee, G. T., 2005. Robots, robots, robots. In *Collaboration@Work, The 2005 Report on New Working Environments and Practices*. European Commission. www.amiatwork.com.
20. McKee, G. T., and Barson, R., 1996. Using the internet to share a robotics laboratory. In *International Journal of Engineering Education*, Vol. 12, No. 2.
21. McKee, G. T., and Schenker, P. S., 2000. Networked robotics. In S*PIE Proceedings on Sensor Fusion and Decentralised Control in Robotics Systems III*, Boston, November, pp. 197–209.
22. Moravec, H. P., 1983. The Stanford Cart and the CMU rover. In *Proceedings of the IEEE*, Vol. 71, No. 7, pp. 872–884.
23. Paredis, C. J. J., and Khosla, P. K., 1993. Synthesis methodology for task based reconfiguration of modular manipulator systems. In *Proceedings of the 6th Int. Symposium on Robotics Research* (ISRR'93).
24. Parker, L. E., 1998. ALLIANCE: An architecture for fault-tolerant multirobot cooperation. In *IEEE Transactions on Robotics and Automation*, Vol. 14, pp. 220–240.
25. Sheridan, T. B., 1992. *Telerobotics, Automation and Human Supervisory Control*, MIT Press.