Journal of Intelligent and Robotic Systems **34:** 1–26, 2002.

A New Conceptual Approach to the Design of Hybrid Control Architecture for Autonomous Mobile Robots

H. YAVUZ
Mustafa Kemal Universitesi, Muhendislik Mimarlik Fakultesi, Makina Muhendisligi, Bolumu 31200 Iskenderun, Turkey

A. BRADSHAW
Engineering Department, Lancaster University, Lancaster LA1 4YR, UK

(Received: 5 February 2001; in final form: 26 June 2001)

Abstract. A detailed analysis and comparison of various control architectures is presented in order to meet the challenging design requirements targeted. All the present advanced control systems have certain advantages and disadvantages compared with each other. Due to the lack of an optimal control system with desired capabilities, such a control system has been the focus of recent robotics research programs. The new approach proposed in this paper is a hybrid control system that takes the advantages of various control structure types thereby integrating them in a way that results in an overall increase in synergy. The proposed control architecture presents a new approach to the design of supervisory control system that utilizes reactive, deliberative, distributed and centralised control approaches, and uses fuzzy logic as well as modular hierarchical structure. The architecture carries out supervision, modification and execution of commands generated by the centralised command arbitration module by conducting fuzzy logic integration of activated behaviours from distributed, independent asynchronous decision making processes that takes information from the user, sensory system and task description, thus providing goal-oriented, real-time responsive and tele-operable control system architecture. The resulting control system was experimented on and it was observed that not only was the response time sufficiently short, but also it exhibited robustness, flexibility, adaptability, portability and expandability.

Key words: mobile robot, control architecture, artificial intelligence, sensor fusion, fuzzy logic, dead-reckoning, conceptual design.

1. Introduction

The design and development of supervisory control systems for autonomous mobile robots continues to challenge researchers. The challenge lies in the development of robust, flexible, adaptive, high-performance control systems that are capable of coping with the dynamics of the real world (Guglielmelli et al., 1997). Many researchers (Brooks and Connel, 1986; Harris and Moore, 1989; Dam and Krose, 1993; Li, 1994a; Dieguez et al., 1995; Zelek and Levine, 1996; Arkin, 1998), to mention a few, recognise the challenge and define the need in a more

Table I. Various reported control systems and their architectural features

Architectural specifications		Brooks and Connel, 1986	Hall et al, 1986	Elfes, 1987	Flynn and Brooks, 1989	Harris and Moore, 1989	Sekiguchi et al, 1989	Arkin, 1990	Freyberger et al, 1990	Choi and Latombe, 1991	Holenstein and Badreddin, 1991	Leonard and Durant-Whyte, 1991	Gasos et al, 1992	Gat, 1992	Heikkila and Roning, 1992	Dam and Krose, 1993	Dorigo and Schnepf, 1993	Kweon et al, 1993	Yen et al, 1993	Ai et al, 1994	Masoud et al, 1994	Chatila, 1994	Li, 1994a	Santos et al, 1994	Tunstel and Jamsihidi, 1994	Aylett et al, 1995	Beaufrere and Zengloul, 1995	Dieguez et al, 1995	Lee, 1995	Palmer, 1995	Tieche et al, 1995	Kurz, 1996	Zelek and Levine, 1996	Guglielmelli et al, 1997	Min et al, 1997	Arkin, 1998	Tubb and Roberts, 1998	Abdelhameed, 1999	Reif and Wang, 1999	Cho and Song, 2000	Ebner and Zell, 2000	Novakovic et al, 2000	Wu et al, 2001	Kasper et al, 2001	Proposed System
Reasoning	Representative (deliberative/MB)		✓	✓		✓	✓	✓	✓			✓	✓	✓	✓	✓						✓		✓	✓	✓	✓	✓	✓	✓		✓	✓		✓	✓									✓
	Reactive (BB/PF)	✓			✓			✓		✓	✓			✓	✓		✓	✓	✓	✓	✓	✓	✓		✓	✓		✓	✓		✓		✓	✓		✓	✓	✓	✓	✓	✓		✓	✓	✓
Control type	Top-down		✓			✓	✓						✓	✓		✓	✓		✓	✓				✓				✓	✓	✓	✓		✓		✓			✓	✓	✓	✓	✓	✓	✓	✓
	Bottom-up																											✓			✓			✓					✓						✓
Processing	Central			✓				✓	✓	✓	✓	✓			✓					✓		✓	✓	✓		✓		✓		✓		✓	✓		✓	✓		✓		✓	✓	✓	✓	✓	✓
	Distributed	✓																													✓			✓		✓			✓						✓
Integration method	Sensor fusion			✓								✓			✓					✓		✓										✓	✓					✓			✓				
	Command arbitration																																										✓	✓	✓
	Behaviour selection	✓			✓		✓		✓								✓	✓				✓	✓			✓											✓		✓						✓
Support AI	Fuzzy Logic					✓							✓	✓					✓				✓		✓		✓		✓							✓	✓					✓		✓	✓
	Neural-Nets						✓									✓								✓				✓										✓		✓				✓	

or less similar manner. Although many agree on the need, the proposed solutions differ widely. The list in Table I identifies the architectural features of a number of reported systems.

Uncertainty plays an important role in many real-time applications (Rosenblatt, 1997) and has been one of the main aspects of study in mobile robotics (Brooks, 1984; Noborio et al., 1990; Mountarlier and Chatila, 1993; Li, 1994b; Wei, 1994; Dam et al., 1996; Min et al., 1997). In this paper the ability to cope with uncertainty is one of the main design constraints. Moreover, prior knowledge of the environment may be incomplete and time variant. Therefore reasoning should be based on up-to-date information of the state of the environment, and should occur rapidly enough to enable an appropriate response to unexpected events. This adaptability function is responsible for coping with the dynamic changes in the environment that occur simultaneously with the operation of the robot.

In an unstructured and dynamic environment the control system must be *reactive* in the sense that its decisions must consider current sensor information and the state of the environment at all times. Therefore, it is essential that the *adaptability function be based on the reactive behaviours* (Yavuz and Bradshaw 1999; Xu, 2000).

Navigation of the mobile robot implies the meaningful progress towards the achievement of a goal. Therefore, the mobile robot architecture must combine *deliberative* or *goal-oriented planning* with *reactive sensor-driven behaviour* in order to be successful in navigational tasks (Yavuz et al., 1999). The obvious need for combining these two architectures has been the reason behind the three-layered control architecture reported by Zelek and Levine (1996) and the alternative architecture presented in this work.

Another important aspect of mobile robot control systems is the need to combine information from several different sources such as information provided by various sensors and task descriptors. Control system architectures also have to be capable of combining objectives such as following paths, avoiding obstacles and reaching goal destinations as well as allowing for tele-operation. To be really useful, the control system must be readily usable on different robots thereby providing *portability*. Hence, the architecture should be able to accommodate and integrate sub-systems that have been developed independently, thereby providing *expandability*. It is also highly desirable to be able to add new sub-systems without disrupting established functionality, thus providing for evolutionary development with built in *flexibility* and *modularity*. In addition, because the various sensors used will, in general, operate at different rates as will the procedures that process their data, they must be allowed to operate *asynchronously* to *maximise the throughput* and thus *the responsiveness* of the system, thereby *maximising the performance of the overall system*.

When dealing with a physical system such as a mobile robot, it is also important to consider aspects of control such as stability and the limitations and constraints of the physical device. The limited capabilities of the actuators, the delays inherent in the system that arise from latencies in data acquisition and processing, inter-module communication delays and the continuous motion of the vehicle may all play a part. This implies that by the time the command is being executed the vehicle is no longer in the current state but actually in a future state. An asynchronous distributed system presents an additional challenge in that, in general, the size of these latencies will be different for each module due to varying processing needs and sensor frame rates. If the various latencies of the system are not accounted for then the vehicle control will be unstable. Since latencies can be taken into account up to a certain level, to be able to stabilise the system, some necessary precautions have to be taken. The distributed timing functionality, built-in pre-processing and Sample and Hold (S&H) facilities in sensor modules are part of the solutions on the sensory system side, aiming at improving the data flow rate. On the controller side on the other hand, the speed of the system has to be fast enough to be able to minimise the time delays. This is achieved by implementing various structural types in the design of various parts of the controller, including *top-down*, *bottom-up*, *central* and *distributed* architectural structures.

When designing software architectures for the control of complex real-time systems such as mobile robots, the architecture must provide the means by which the system may accomplish its objectives efficiently. It must be able to satisfy real-time constraints, promote fault tolerance and provide for the safety of the vehicle and its surroundings. Moreover, crucial consideration in the design of mobile robot architectures is the ease with which a system may be developed, tested, debugged and understood.

It is clear that the issues involved in the selection of the architecture for the control of autonomous systems is a highly complex task that requires the consideration of the task requirements along with the consideration of uncertainty and real-time responsiveness. In this paper, an architectural framework is proposed that has been demonstrated to satisfy these essential constraints. The following statement describes the proposed architecture and sets the goal for the development of a novel control architecture:

> *Supervision, modification and execution of commands generated by the centralised command arbitration module using fuzzy logic integration of activated behaviours from distributed, independent, asynchronous decision making processes are carried out by processing sensory system information, user command input and task information, to provide goal-oriented, real-time responsive and tele-operable control system architectures. Furthermore, the framework proposed in developing and integrating distributed, independent decision making modules communicating with such a command arbiter leads to the evolutionary creation of robust systems with greater capabilities, thereby allowing modular, flexible, portable, adaptable and robust supervisory control systems design.*

2. Control Architectures

2.1. REACTIVE OR DELIBERATIVE REASONING

In deliberative systems, actions are generally obtained by a sequence of planned processes (see Figure 1(a)). Therefore, achievement of the desired goal-state from the initial state is either by the previously prescribed steps, or by steps determined as a result of an optimal path search (Dieguez et al., 1995). Upon determination of the step to be performed, the necessary control command is generated and then handed off to a run-time execution module. Thus, the operating principle of these systems is the generation of a world model, search and generation of a plan, and then execution of the planned steps one by one. It is clear that these systems suffer from severe drawbacks the chief of which is slowness due to the combinatorial explosion of the search process in optimal path planning algorithms (Aylett et al., 1995). To perform such a search for the optimal path, the robot would need to stop and deliberate for a while before performing the next motion as, for example by Shakey (Arkin, 1998). The delay would deteriorate the performance of the system, especially if the search algorithm is a computationally heavy one (Aylett et al., 1995). Furthermore, typically no sensing or planning takes place during the execution of a step, and the effects of an action are assumed to have

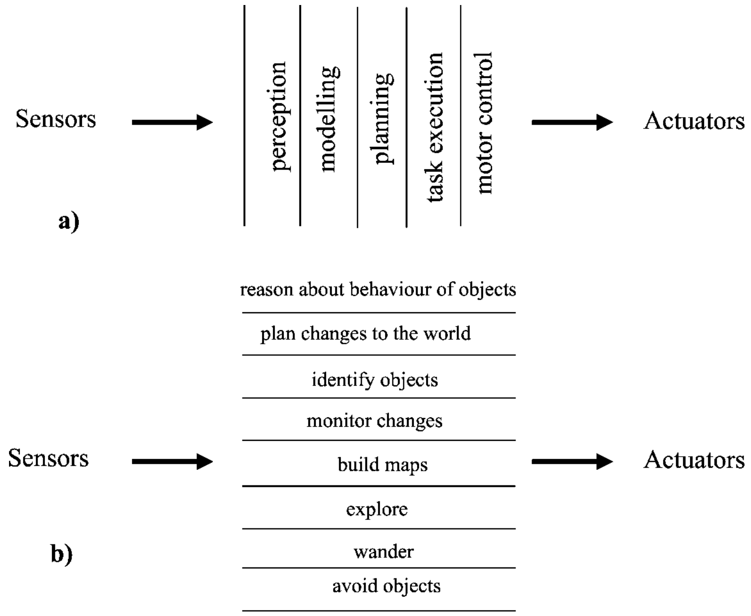

Figure 1. Decomposition of mobile robot control systems: (a) conventional systems, (b) Behaviour-Based (BB) system (Brooks and Connel, 1986).

occurred without verification, thus worsening the robot's unresponsiveness to the environment (Rosenblatt, 1997).

Reactive systems on the other hand, are designed to respond directly to external stimuli. They tend to use compiled procedural knowledge instantaneously to map perceptions to actions using the sensory information (Brooks and Connel, 1986; Kasper et al., 2001). However, the drawback of such systems is that a typical reactive system and its reactive execution perform no look-ahead and explicit evaluation of possible future states but only the current state of sensory information (see Figure 1(b)). Hence, the distributed direct-perception-action feature might result in inefficient actions and failure to achieve the goal (Aylett et al., 1995; Kasper et al., 2001).

One of the problems that deliberative systems suffer severely from is the uncertainty in perception. However, reactive systems simply avoid it by sensing at a rate high enough so that false readings have a very limited impact, thereby presenting a simple but effective solution. Therefore, problems with uncertainty in action are tackled by acting on the currently perceived world at all times. Besides, no assumptions are made about the persistence of previously observed states, and world models and knowledge based planning are avoided under the belief that "*the world is its own best model*" (Brooks, 1991a).

Nevertheless, by disallowing the internal representations, such systems not only raise the requirement for making environmental features of interest visible to sensors and behaviour modules at all times, but also disallow benefiting from consecutive perceptions, thus adding unnecessary constraints and reducing the flexibility of the overall system. Hence, without internal representations and plans, there is no means to direct the robot towards a goal unless the clues are directly observable. To compensate for this weakness reactive systems usually wander around, essentially performing a random search, until they happen to reach a point where the means of achieving the goal becomes apparent (Brooks and Flynn, 1989; Nehmzow and McConigle, 1993; Baroni and Fogli, 2000).

The conclusion on the architectural reasoning is that; a robot control system should balance the need for carefully planned, theoretically optimal actions to achieve goals against the need for quickly decided, satisfying reactions to a dynamic, uncertain environment. Deliberative reasoning provides the ability to achieve high-level goals, while reactive responses enable the robot to adapt and respond to the dynamic world. The absence of either type of reasoning could result in the robot being unable to satisfy the objectives of being both goal-oriented and adaptive (Chatila, 1994; Aylett et al., 1995).

2.2. DISTRIBUTED OR CENTRALISED PROCESSING

As with any complex system, trade-offs must be made between the straightforwardness of a centralised system on one hand and the responsiveness, robustness, and flexibility of a distributed system on the other. In centralised architectures, all sensor data must be collected and fused into a single world model as shown in Figure 2. The use of a "blackboard" where all the information is gathered from various sources and a control command is generated centrally is often used (Liscano et al., 1995). In such systems a complete path considering all relevant information is planned within this world model and constructed on this "blackboard". Once the planning process is started, the system disregards any other operation or necessary action to be taken. Until the planning sequence and control commands are generated, it disregards all other duties or actions, no matter how simple or urgent that duty or action might be. Such systems contain no parallelism or pipelining. Furthermore, once a plan sequence is formulated, it is followed in an open-loop fashion while disregarding the dynamic changes and consequent differences between the environment and the world model. No mechanism is provided to perceive if commanded actions actually have the desired effect or not. In addition to introducing potentially harmful delays due to its open-loop operation a centralised architecture also leads to brittleness as the system may fail entirely if any single subsystem fails to function properly. Thus, any modifications or additions made to a system function could cause serious problems thereby disallowing flexibility. To overcome such limitations inherent in purely centralised architectures, schemes for allowing distributed processing were developed.

There are two main types of distributed systems: hierarchical and behaviour based systems. In hierarchical architectures, modules are organised into multiple control levels that operate at varying granularities, levels of abstraction and time

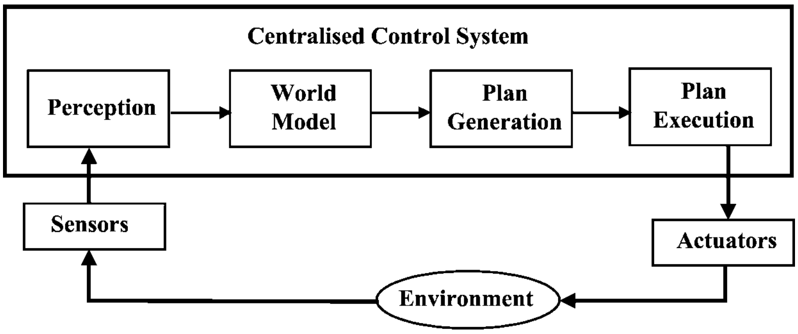

Figure 2. Traditional functional decomposition of centralised systems.

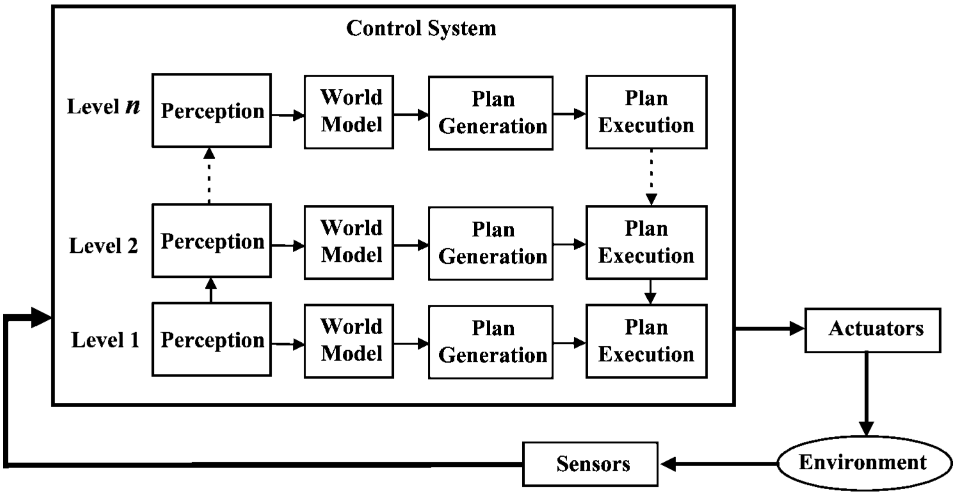

Figure 3. Hierarchical architecture with recursive functional decomposition.

scales. They, therefore, provide varying trade-offs between long-term correctness and completeness and short-term survival and relevance.

Traditional hierarchical architectures use a homogeneous functional decomposition, with each level constructed of the same modules, even at different levels of reasoning (Albus et al., 1987; Rosenblatt, 1997; Arkin, 1998). They are composed of multiple control levels, each of which has the same type of structure as a centralised system, as shown in Figure 3. However, each level operates in parallel at a different rate, so that the lowest levels are free to respond to immediate stimuli without having to wait for higher level reasoning processes.

Behaviour-Based (BB) architectures, on the other hand, constitute a radically different class of robot control systems. Rather than constructing the system of functional modules such as perception and planning, the system is composed of individual behaviours, each performing a specific, limited task (Kasper et al., 2001). This allows the robotic system to be developed in a bottom-up, evolutionary man-

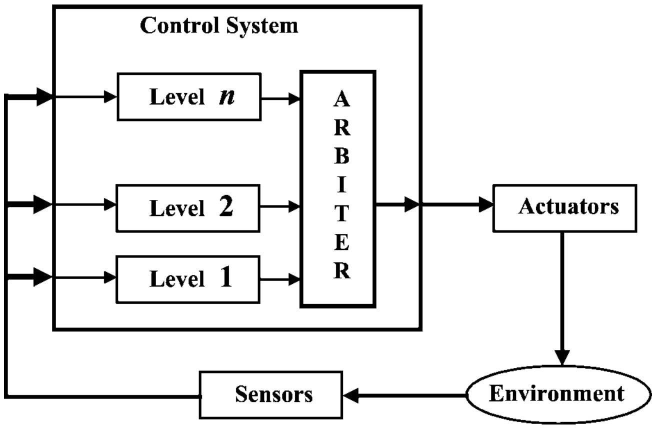

Figure 4. Levels of competence in BB architecture.

ner because behaviours are essentially self-contained and capable of achieving their specific purpose whether or not other particular modules have been developed and are present in the system (Brooks and Connel, 1986). Behaviours encapsulate the perception, planning and task execution capabilities necessary to achieve one specific aspect of robot control. Thus, in such architectures, not only the processing but also the actual control of the robot itself is distributed across multiple independent modules (Rosenblatt, 1997). BB architecture is more robust than other types of architectures because if any one part of the system fails, the rest of the system continues to function independently (Brooks and Connel, 1986; Flynn and Brooks, 1989). The drawback of such a system is that predicting its behaviour is problematic; the interactions between large numbers of asynchronous independent modules are difficult to formulise, analyse and debug (Rosenblatt, 1997). Because an individual component in a BB architecture is capable of producing meaningful action, behaviours can be composed to form levels of competence (Brooks, 1986) as shown in Figure 4.

2.3. COMMAND ARBITRATION AND SENSOR FUSION

Architectures can also be characterised according to the methods using which they combine information and objectives. Architectures that perform sensor fusion construct a single unified world model based on all available sensory data (Leonard and Durrant-Whyte, 1991; Heikkila and Roning, 1992; Chatila, 1994; Kurz, 1996). The generated world model is then used for planning actions, as illustrated in Figure 5. It is clear that the method allows comparison of the various sensory information and thereby their validation during the generation of the model. Nevertheless, it has the disadvantage of creating a computationally expensive sensory bottleneck in that all sensor data must be collected, evaluated, validated and integrated before it can be acted upon. Another difficulty is actually integrating the information from various

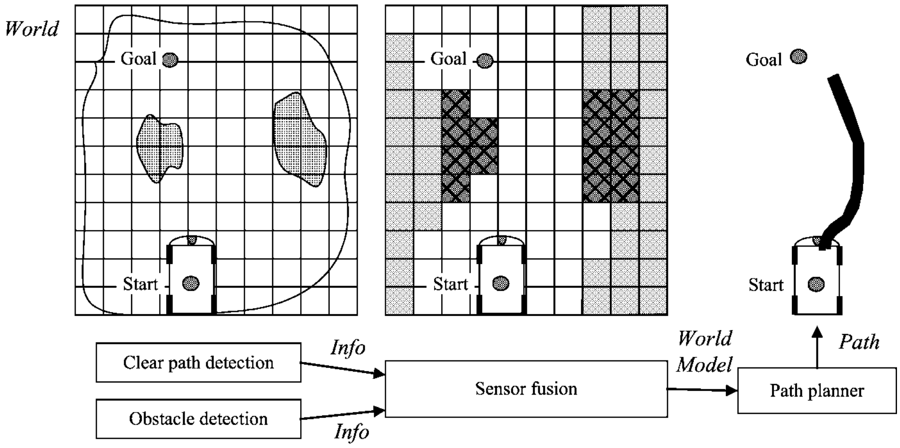

Figure 5. Sensor fusion and world model for planning.

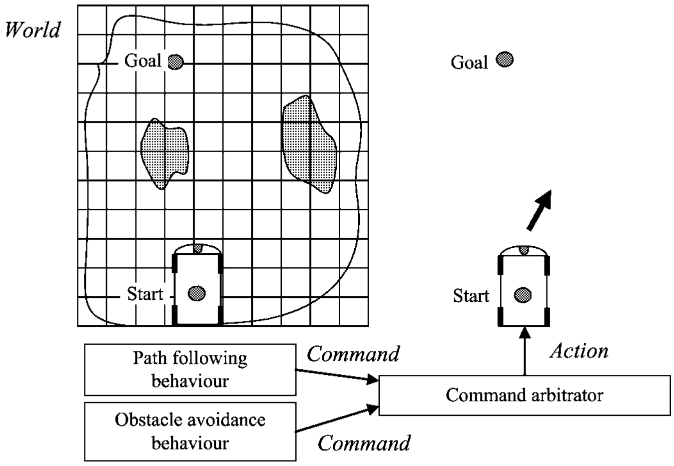

Figure 6. Command arbitration and integration of behaviour commands.

sources, as they may not be suitable for representation in a single map or model. Thus, by requiring a single representation for all sensor and map data, a centralised architecture does not allow specialised modules to use other representations and algorithms that might be better suited to the task.

In contrast, BB architectures do not create a central world model; instead, the perceptual processing is distributed across multiple independent modules (Brooks and Connel, 1986; Aytell, 1995). Each behaviour requires only fragmentary knowledge of the world and receives exclusively that sensory data which is directly relevant to its particular decision-making needs, thus each behaviour is free to use whichever representation is deemed most appropriate for the purpose, and there is no need to fuse all available data into a single world model before any module can respond to the processed data.

A behaviour module is a self-contained control block that maps input data into actions as output. However, input data and that action are only concerned with one specific aspect of controlling the robot to achieve its tasks. Therefore, in a BB system, it is necessary to select among or combine the actions suggested by the various modules to produce an action that meets the needs of the overall system. This is the role of the command arbitration module shown in Figure 6. By appropriately combining behaviour commands through arbitration, a robot control system can respond to its environment without suffering the problems inherent in sensor fusion. However, command arbitration runs the risk of losing information valuable to the decision-making process. Therefore, a careful balance must be achieved between completeness and optimality of fusion, and modularity and efficiency of arbitration type of systems.

2.4. TOP-DOWN AND BOTTOM-UP CONTROL

In hierarchical architectures, the design and operation of robot control systems are expected to proceed in a top-down manner; each level controls the level beneath it and assumes that its commands will be executed as anticipated. However, expectations are not always met, and there is a need to monitor the progress of desired actions and to report failures as they occur. In unstructured dynamic environments, this approach introduces complexities and inefficiencies. These could be avoided if higher level modules that participate in the decision-making process do not assume that their commands will be strictly followed. A solution to the problem could be the activation of a set of behaviours instead of just one. In such architectures, an event driven higher level module enables and disables groups of behaviours at the lower levels according to the currently appropriate mode of operation and information, thus allowing several behaviours to operate at once.

In contrast to hierarchical architectures, all behaviours in the BB architecture are active at all times, in the sense that they receive and process data on a continuous basis (Brooks and Connel, 1986). The control structure is bottom-up, since each behaviour itself determines if it is relevant to the current situation, making the judgement on the data that it receives. For example, a behaviour designed to detect and avoid obstacles based upon sonar data would continually read the data to determine if there are obstacles in the robot's path; when obstacles are detected the behaviour would issue avoidance commands, otherwise it would remain non-active.

3. A Hybrid Architecture

The review of the architectural features for supervisory control systems has revealed that there is a need for combining some of the features together to form a novel robust flexible modular control system. Therefore, it is necessary to assess each of the system features against the requirements, to guide the selection and

Architectural Specifications \ Requirements		Uncertainty	Reflexiveness	Ease of application	Flexible and modular	Fast response	Optimal operation	Task learning	Goal-oriented	Planning	Robustness	Efficiency	Completeness	Multi-mode operation
Reasoning	Representative (deliberative/MB)	▲	▲	▲	▲	▲	⊙	⊙	⊙	⊙	△	△	◯	◯
	Reactive (BB/PF)	⊙	⊙	⊙	⊙	⊙	▲	○	△	△	⊙	⊙	△	○
Control type	Top-down	△	△	△	△	△	⊙	⊙	⊙	⊙	△	△	◯	◯
	Bottom-up	⊙	⊙	⊙	⊙	⊙	△	△	△	△	⊙	⊙	△	○
Processing	Central	△	△	△	○	○	⊙	◯	◯	◯	△	△	⊙	◯
	Distributed	⊙	⊙	⊙	◯	⊙	△	△	△	△	⊙	⊙	△	△
Integration method	Sensor fusion	△	△	△	△	△	◯	▲	◯	◯	▲	▲	⊙	○
	Command arbitration	◯	◯	◯	◯	◯	⊙	▲	◯	◯	◯	◯	○	⊙
	Behaviour selection	⊙	⊙	◯	⊙	⊙	○	▲	▲	▲	⊙	⊙	▲	▲

Legend ← Satisfaction improves

⊙ Very good ◯ Good ○ Moderate △ Not good ▲ Very bad

Figure 7. QFD analysis of the control systems architecture specifications vs. requirements.

unification of architectural features. For this purpose, a quality management tool called Quality Functional Deployment (QFD) is used. The results of the analysis are presented in Figure 7.

The analysis presented in Figure 7 shows the need for integration of various features to achieve the best performing supervisory control architecture. It is also clear that both types of reasoning are necessary to achieve a goal-oriented and reactive system. Therefore, a deliberative upper module is needed to make high-level navigational planning and decision-making, whereas a low-level reactive module is needed to provide the reflexive and responsive part of the controller.

The high-level navigational planning tasks are carried out as by sensor fusion, map generation and then path planning. Afterwards these optimum commands are sent to lower levels. Such an approach has disadvantages and difficulties as mentioned in previous sections and in Figure 7. The solution is to avoid using any kind of complex maps, thereby eliminating the need for sensory fusion and all the other related computationally heavy processes. Instead, the solution proposed in this work is to use a simple sequential task-information set that only contains simple steps of a navigational pattern that direct the robot towards the goal or the achievement of the task pattern. This is done by learning through a teach mode of operation, during which the robot samples and records the task pattern, thereby providing the solution to the task information source problem.

The reactive low-level module, on the other hand, provides the distributed processing mechanism through which the overall sensory information processing is carried out by reactive modules. These reactive modules receive the information and consequently, generate control settings reflecting a particular part of the reflexive-behaviour control-settings information, which is to be used in the generation of the control command.

Another requirement for the control system architecture of the robot is to allow tele-operation. Thus, the robot is required to recognise the operation mode and activate the necessary modules to provide the necessary functionality. Moreover in some cases, whether in teach mode of operation or in playback mode of operation, there might be cases when the user might wish to interrupt the robot's actions. Handling such user inputs in teach mode would not cause any problems or complicate the control systems design as the user inputs are the only source of control commands being evaluated and acted upon. However, in playback mode the case is different, and the operation in this mode requires the integration of deliberative and reactive module inputs. The addition of user input to the system further complicates the overall control system command integration module, unless necessary precautions are taken. For this purpose, in playback mode of operation, the robot evaluates the inputs from the user, deliberative and reactive modules, and assigns priorities to the requested actions.

At this stage, it is clear that there is a need for a mechanism to combine the information from various sources and achieve a high performance motion control. Since inputs from various sources are in different formats (task plan step, reflexive behaviours and control commands), there is a need to convert all these formats into one before the integration process. Therefore, each input set is firstly evaluated and then decoded/converted into standard behaviours with specific settings used in both the motion control and activation systems. Then each set is analysed, sorted and sent to the command arbitrator, where the integration of sets of information and generation of the control command is achieved. To compensate for the delays and resulting control problems, the control command is checked against the sensory information before the execution. This in turn allows the high-priority reflexive-control-action requests to bypass the control command process and attend to the urgent reflexive behaviour control requests. The proposed architecture is shown in Figure 8, where the features of the model are labelled for identification and visualisation.

By converting all information into behaviours, the dynamics of the vehicle are taken into account. In addition, by the integration of sets of information from various sources, a goal-oriented, reactive and tele-operable control system is achieved.

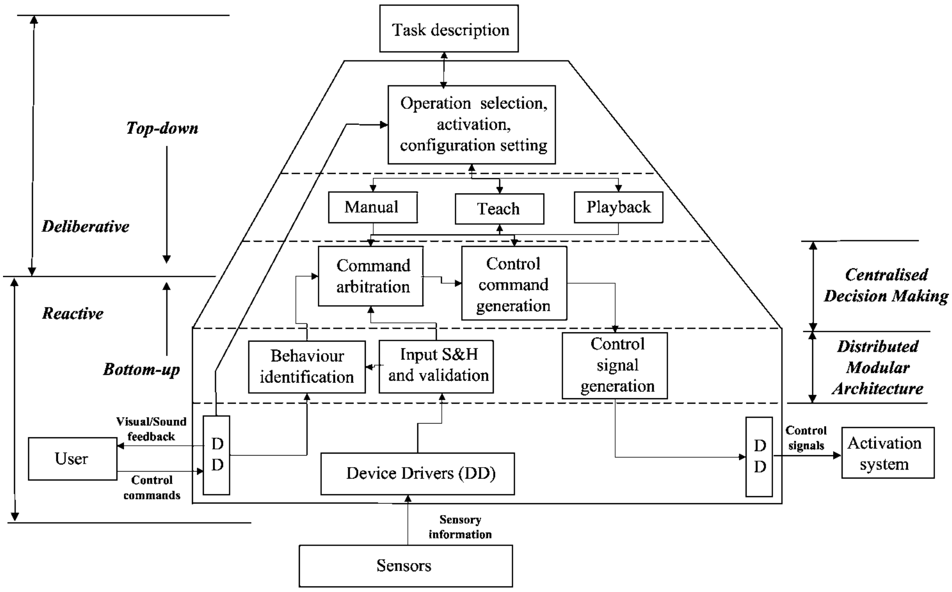

Figure 8. Composite control architecture for an intelligent autonomous mobile robot.

With the introduction of modularity and the information processing technique into the design, portability and expandability features are achieved. Besides, the modular reactive design provides the flexibility that allows design level modifications and future alterations. Another advantage of the reactive design is that synchronisation related problems are eliminated, thereby providing maximum throughput and responsiveness, thus maximising the performance of the overall system. The introduction of the bypass technique further contributes to the performance improvement, minimising response time delays. The overall sensory processing is distributed to low-level reactive modules, whereas the central modules benefiting from both of the processing techniques carry out the main decision making issues. Thus, the developed system allows sensor specific processing in reactive modules bringing the responsiveness, robustness and flexibility to lower-levels, and main decision making in central modules for straightforwardness, modularity and efficiency. The operation of deliberative modules is achieved top-down, and the reactive modules bottom-up, which is quite typical in such systems.

4. Details of the Proposed Control System Architecture

The specifications for the control architecture set in Section 2 imply that the mobile robot requires not only tele-operation, but also self-supervised goal oriented autonomous operation. The former could be achieved as manual mode of operation. The latter, however, is accomplished firstly by transferring the task and goal information to the robot, and then executing the task information in an autonomous manner, thereby achieving teach and playback modes of operation, respectively.

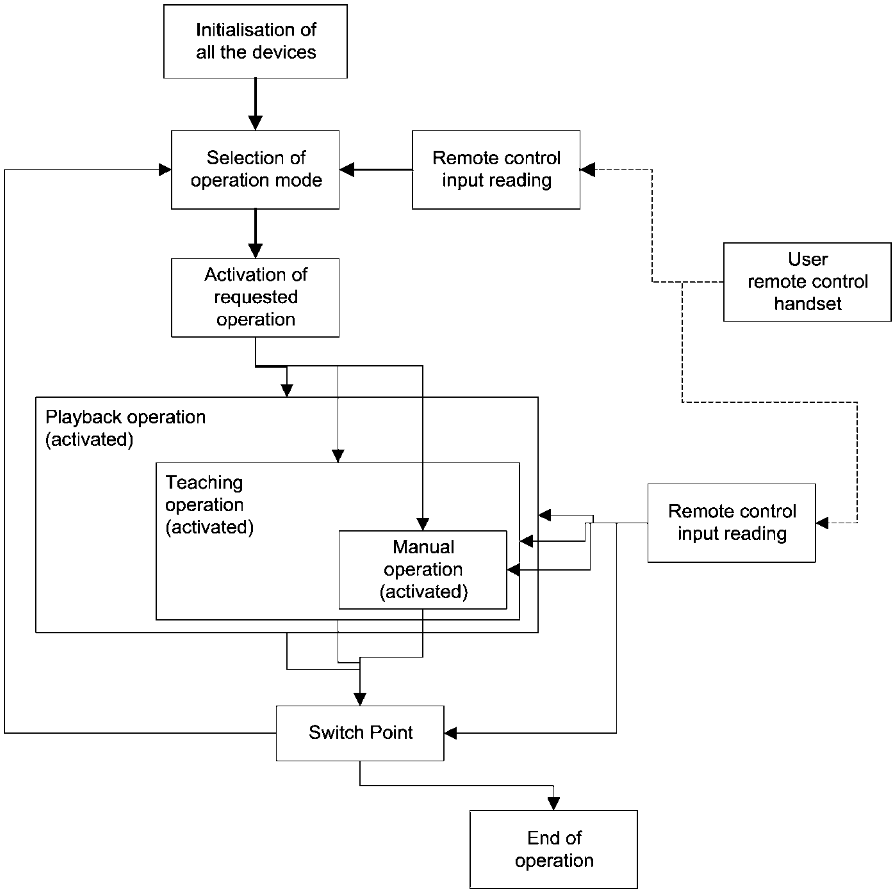

Figure 9. Activating and switching operation modes.

Therefore, there is a need for a teach session, i.e., information transfer, before performance of a new task during which the robot is required to be manually operated and also to be recording timed used commands in a systematic manner. The playback mode is then performed by executing the taught session information under self-supervision functions in an autonomous manner with the support of adaptability, decision-making mechanisms and other related functions.

Considering the functionality required for the operation modes, one can observe that they have common functions although they differ from each other as a whole. In fact, as illustrated in Figure 9, the manual mode is a subset of the teach mode operation, and the teach mode operation itself is a subset of the playback mode. The integration of the overall system requires integration of the subsystems that provide the manual mode of operation as the first stage. Then, by the addition of the necessary teaching related subsystems to the manual robot, the overall system could be provided with learning ability (Brooks, 1991b), thus achieving the teach mode of operation. However, it is the playback mode of operation that requires the robot to operate intelligently and autonomously by making decisions and adapting itself to the surrounding dynamic work space. Because the integration of sensing, intelligent decision making, reasoning functions and related subsystems are achieved at this phase of the design, it is treated with particular care and attention to detail.

Figure 9 illustrates the operation principle of the robot and the mechanism that allows sequential continuous operation. To operate the robot, the user switches the power on. Then, before the execution of any command, the software initialises all the hardware devices available on-board. Afterwards, the user is required to choose the desired operation by pressing the assigned remote controller buttons. Depending on the request, a set of operational functions is activated, either manual, teach or playback. In manual mode; only manual, in teach mode; only manual and teach, and in playback mode; manual, teach and playback operation functions are activated.

In the teach mode of operation, the control command sample and record function was designed only to store information on the command type and period in which it was active. This information could then be processed to yield point-to-point navigation using the dead reckoning functionality. In Table II, the manoeuvres with related control parameters and equations governing the motion are listed. The specification of manoeuvres listed in the table is used in designing the dead-reckoning system that is based on manoeuvres/behaviours. The advantage of the system is that the executed behaviours or manoeuvres could be directly used in position determination. Besides, due to the modular design principle, two separate dead-reckoning modules could be run in parallel, providing positioning information and the difference in localisation for both paths. This is an important feature, as the adaptability feature requires compensation on modification and a return to the original path prescribed by the task information source.

Another important issue in the dead-reckoning module design is to analyse the motion of the robot and to see if it fits the criteria set by the kinematic analysis before making a positioning update. For this purpose, heading-sensing information is provided to the dead-reckoning module through which the performances of individual behaviours are inspected. The behaviours numbered 1 and 2 in Table II are expected to be executed without a change in the heading, whereas the others with a certain heading change which is set by the control parameter ω_{B}. Thus, the observed heading change is included in the computation of positioning, thereby providing compensation for slippage and other problems that may be encountered.

Having defined and designed the dead-reckoning module, further integration of the intelligent robot could be achieved by reading the task information, and converting the information into point-to-point navigation information, and then executing the necessary commands to achieve such a motion pattern. However, there is a need for a module to receive the information from the dead-reckoning module and convert it into timed motion control commands in the form of behaviours.

Table II. The Manoeuvres and motion control parameters

No	Manoeuvre	Motion control parameters					
		ω_L	ω_R	ω_B	V_L	V_R	V_B
1	Forwards	ω	ω	0	ωR	ωR	ωR
2	Backwards	$-\omega$	$-\omega$	0	$-\omega R$	$-\omega R$	$-\omega R$
3	CCW rotation	$-\omega$	ω	$\frac{2\omega R}{l_a}$	$-\omega R$	ωR	0
4	CW rotation	ω	$-\omega$	$-\frac{2\omega R}{l_a}$	ωR	$-\omega R$	0
5	Forwards left cornering	$\frac{\omega}{2}$	ω	$\frac{\omega R}{2l_a}$	$\frac{\omega R}{2}$	ωR	$\frac{3\omega R}{2}$
6	Forward right cornering	ω	$\frac{\omega}{2}$	$-\frac{\omega R}{2l_a}$	ωR	$\frac{\omega R}{2}$	$\frac{3\omega R}{2}$
7	Backwards left cornering	$-\frac{\omega}{2}$	$-\omega$	$-\frac{\omega R}{2l_a}$	$-\frac{\omega R}{2}$	$-\omega R$	$-\frac{3\omega R}{2}$
8	Backwards right cornering	$-\omega$	$-\frac{\omega}{2}$	$\frac{\omega R}{2l_a}$	$-\omega R$	$-\frac{\omega R}{2}$	$-\frac{3\omega R}{2}$
9	Forwards left U-turn	0	ω	$\frac{\omega R}{l_a}$	0	ωR	$\frac{\omega R}{2}$
10	Forwards right U-turn	ω	0	$-\frac{\omega R}{l_a}$	ωR	0	$\frac{\omega R}{2}$
11	Backwards left U-turn	0	$-\omega$	$-\frac{\omega R}{2}$	0	$-\omega R$	$-\frac{\omega R}{2}$
12	Backwards right U-turn	$-\omega$	0	$\frac{\omega R}{l_a}$	$-\omega R$	0	$-\frac{\omega R}{2}$

V_R: speed of right power wheel, V_L: speed of left power wheel, V_B: speed of robot body, ω_R: angular speed of right wheel, ω_L: angular speed of left wheel, ω_B: angular speed of robot body, ω: angular speed, R: wheel radius, l_a: distance between wheel centres.

By integrating playback control, sensory system input functionality and decision making capability, the robot is provided with the grounds for the adaptability. To achieve the adaptability function, the sensory information, task information and user inputs and their analysis results are used in the decision making process. For this purpose, instead of only looking at the collision sensing information, as for manual and teach mode operations, the sensory information set is used as a whole. However, use of the sensory information requires processing of raw sensory information in the software based Device Drivers (DD) and hardware based Device Interface Units (DIUs) where all sensory information is evaluated, and behaviour-oriented motion control command is generated. Due to the differences in individual needs and analysis criteria of each sensor array, the processing modules are provided with specific sensor-information-analysis modules that are embedded in the relevant DDs and DIUs.

The second set of information input to the main controller is the task information details discussed in the previous sections. However, the task information evaluation and integration to the system is modified as necessary to allow the integration of all three sets of information to the controller, combining the user, sensor and task information, and to generate a control command accordingly. The third set of information is the user information. It mainly serves the purpose of assistance in stages where the robot's intelligent level supervisory controller fails

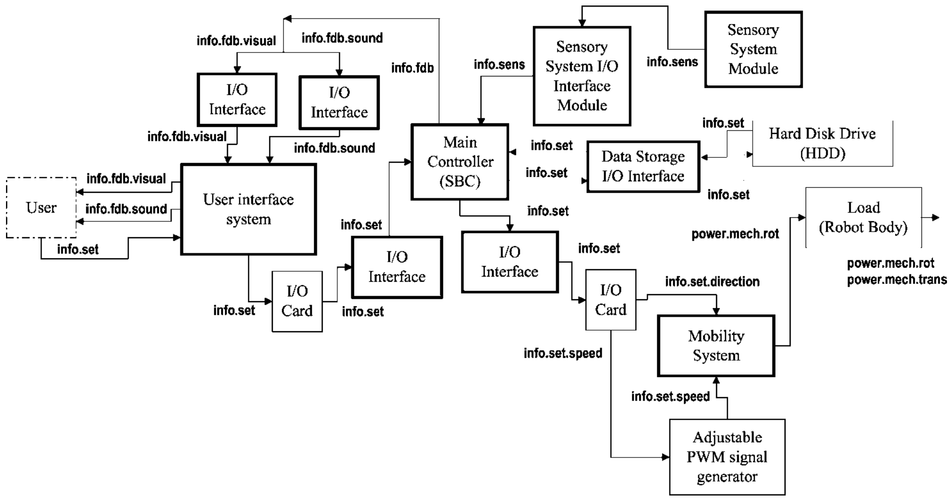

Figure 10. Overall control system model.

to produce the necessary motion to overcome the complications of the dynamic environment. To combine the information sets received from these sources, a fuzzy logic based decision-making mechanism is proposed. The decision-making uses a set of rules through which the input space is analysed and fuzzified. Then, by using the behaviour-categorisation-sorting and central-command-arbitration modules, the control command is generated. For this purpose, the Sugeno Fuzzy Logic (FL) technique is used in the defuzzification process in the command arbitration module design.

The Figure 10 illustrates the overall system structure including the main controller (Single Board Computer – SBC) module and the other system modules used in the design of the control system architecture. As can be seen from the block diagram model in the figure, all subsystems in the architecture are designed and built to be modular. Besides, the inter-module communications are pre-defined with prescribed input output signal ports as labelled. In addition, the data acquisition amongst hardware and software systems are achieved using I/O interface modules that are composed of software based Device Drivers (DDs) and hardware based Device Interface Units (DIUs).

One of the unique features of the above proposed control system architecture is that each is a separate subsystem and a module that could be modified or replaced as necessary. This feature allows the control architecture to be transferred on to a new robot platform provided that necessary modifications are made, thereby providing the portability, modularity and flexibility aimed at.

Although the control system architecture provides solutions to the problems discussed in Section 2 and 3, the most unique and vitally important module of all is the main controller module where central processing, decision making and

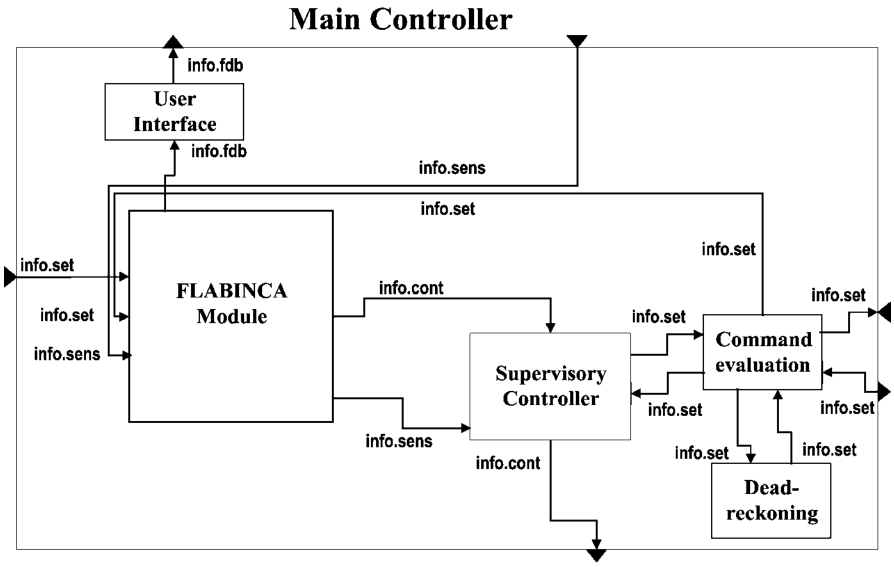

Figure 11. Main controller module.

command generation tasks are achieved. The module, as shown in Figure 10, takes and sends three inputs signals (user command, sensory system and task information) and three output signals (user interface, control command and task information), respectively. A detailed illustration of the main controller module is given in Figure 11.

Figure 11 illustrates the main controller where the control signal generation, control command evaluation and supervision as well as dead-reckoning and user interface signal generation tasks are achieved. The main controller module includes *supervisory control module, dead-reckoning, user-interface* and *command evaluation modules*. In the figure, the inter-module signal port types and the signal-types and the corresponding physical parameters (V_L, V_R) are also illustrated in detail. The command evaluation and supervision modules inspect the operation of the controller and take necessary actions if system were to perform an illegal action such as movement towards collision with a detected object. The user interface signal generator module, on the other hand, produces signals for informing the user on the actions being taken. The FLABINCA module is responsible for integrating the user, task and sensory information and generating the control command accordingly.

Figure 12 illustrates the FL-based behaviour analysis-integration and command arbitration (FLABINCA) module. The module consists of two main sections labelled as fuzzification and de-fuzzification modules, where decision-making in behaviour integration and command generation tasks are achieved, respectively. In the fuzzification part of FLABINCA, firstly the input information sets are evaluated individually where active behaviours are identified, prioritised and utility

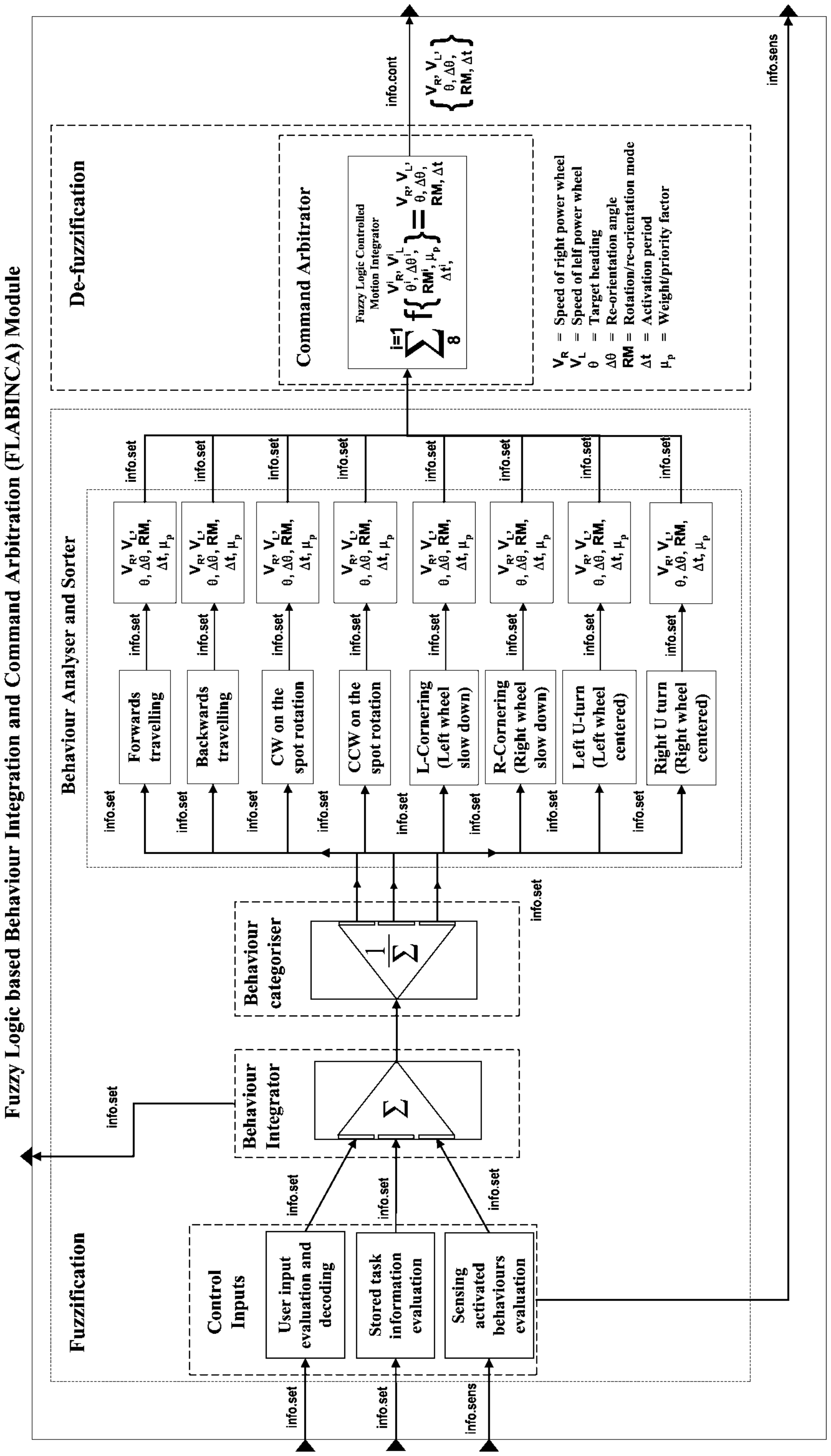

Figure 12. FLABINCA module.

parameters defined. Afterwards, all three sets of information are fed forward to the behaviour integration module where behaviours from the same category are combined by grouping them. In this module, analysis of the behaviours is carried out. In the analysis, the priorities and utility status information is used in identification of behaviours to be activated and eliminated. Then, in the behaviour categorisation module they are sorted according to their behaviour type, and then sent to the behaviour analysis and sort module. In this module, the behaviours are converted into motion control parameters, which are then sent to the control arbitration module where they are used in control command generation. The unique feature of the FLABINCA module is that it integrates information from various sources in the form of behaviours. After the integration process, the activated simple behaviours are then categorised and then combined to form the complex behaviour to be executed. In this way not only are the information sets fused in an effective manner but also the control command to achieve the complex behaviour is generated.

5. Experimental Results

As illustrated in Figure 13, the main use of the FLABINCA module is that it provides the robot with the capability of avoiding obstacles as a part of the adaptability function. The obstacle avoidance function is achieved through the use of reactive behaviours set to active by the sensors that perceive the object blocking the path. The activation of the reactive behaviours by sensors provides the necessary input to the controller to achieve adaptation.

An important aspect however, is to bring the robot back to its original path after the completion of obstacle avoidance. For this purpose, the dead reckoning modules calculations are used to direct the robot towards the original path. In

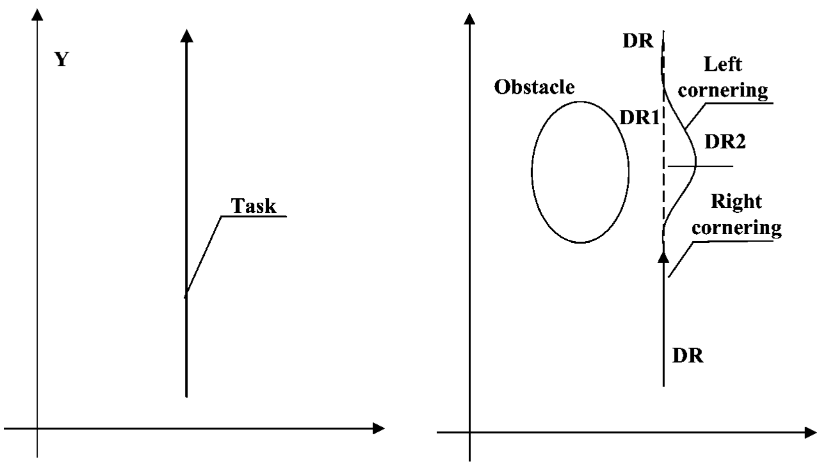

Figure 13. Obstacle avoidance through complex behaviour generation and execution (DR – Dead Reckoning data set).

calculation of the compensation related parameters the subtraction function of the coordinate class of the dead-reckoning function is used.

As a part of the experiments for ensuring system performance independencies amongst integrated subsystems, integrated modules and individual modules are tested for various module or subsystem failures and their effect on overall system performance. The main aim of the tests discussed is to ensure that the designed control system architecture meets the requirements. For this purpose, various tests are carried out including disabling a part of the sensory system to observe the expected robustness of the main controller or purposefully introducing disturbances to the controller to observe the effects of the external stimuli.

The first tests on the complete system are performed by initially manually operating the robot. For this purpose, the control system and the related interface modules are bypassed by setting the manual operation module active. During these experiments, the manoeuvres are tested to ensure the precise control of the power wheel activations. The robot is, then, commanded to follow a straight line of 15 m and the resulting misalignment is measured to be about 7 cm. Then, 24 on the spot rotations are performed and the centre of the robot's misalignment is measured to be about 2 cm. From this experiment, it is concluded that motion control system performed well, wheel slippage is minimal and the resulting positioning error is well below the expected levels. As the manual operation mode disregards the sensory information and the task information, it is also demonstrated that any malfunctioning in these systems has no effect on the system performance.

After the manual operation tests, the teach mode operations tests are carried out to test the systems learning function. For this purpose, the robot is run in manual mode again but this time with the aid of sensory system and the leaning function that uses Learn-From-Demonstration (LFD) methodology. When running in this mode, sensory system information and the user commands are considered. The user commands are performed while watching out for obstacles and warning the user if any detected. The tests performed for this mode of operations has proven that the limitations of the robot's learning capabilities are the limitations prescribed by capacities of the data storage utility (hard disk drive) and the power source (battery). Besides, the malfunctioning of any set of sensory system is reflected to the performance of the system as blindness to a certain obstacles within a certain range, which has not actually deteriorate the performance as control command arbitrator programmed to disregard the malfunctioning sensory system modules. As a result of these experiments it is concluded that the robot carried out the tasks and obeyed the user commands when performing the manual and taught operations.

In the final and most vital part of the experiments the playback mode of operation is tested in which the robot is commanded to perform a straight path following teach operation followed by an autonomous playback of the taught session. In the playback operation, firstly the robot is tested by repeating the taught session in the initial environmental conditions. The playback operation was observed to be reasonably successful with accurately timed motion generation as expected. In the following playback session, changed dynamic circumstances are provided to test the adaptability part of the controller by simply adding an obstacle to block the robot's path (Figure 13). The results of the experiments performed in these conditions indicated that the robot is capable of avoiding obstacles if there is an achievable path between the current and target locations. This restriction is due to a lack of a global-planning facility in the main control algorithm. Although this appears to be a drawback, the experimental results indicate that the exclusion of global planning pays back with real-time applicability and reactivity of the overall system.

The tests performed by disabling certain modules (ultrasonic, IR and touch sensors) of the sensory system revealed that the malfunctioning of those sensors degrades the performance of the system; however, it does not stop it from functioning. The tests on disabling individual sensory system modules revealed that ultrasonic sensors had the least effect on the degradation of the systems performance in adaptability and recognition of changes in the dynamic environment. Disabling IR sensors on the other hand, worsens the performance of the system, as unreliable readings of the ultrasonic sensors are not sufficient to guide the robot. Thus, the unreliable readings from those sensors mislead the controller and degraded the performance of the adaptability function. Whereas disabling the touch sensors made it impossible to detect collision. Therefore, the obstacle avoidance behaviour is not performed properly unless the objects are recognised and avoided long before the robot comes too close to them.

It was also observed during the experiments that the designed dead-reckoning module was able to perform its function of providing positioning information only if the operation of the robot is kept short enough so that the errors accumulated during the operation of the robot, as a result of power wheel slippage, would not misguide the robot. The precision mechanical design of the system supported with accurate motion generation are observed to have improved the performance of the system by extending the period of time the robot stayed active and suffered only small positioning errors. However, it is also noted that there is an obvious need for an external device for positioning updates, which would improve the system performance by extending its operation time to the limits of its power supply.

6. Conclusions

The stage by stage testing of the system and its subsystems proved useful in the achievement of overall system performance maximisation while minimising the problems related to the integration of subsystems and their performances.

The modular, distributed and BB structure of the main control system and the modularity of the overall system not only allowed testing for malfunctioning of various subsystems and observation of the system performance, but also provided the controller with the robustness to cope with such cases and maybe with degraded performance but continuous operation. The experiments carried out on the design of the system proved that modular design of the system eased testing for performance by allowing individual and integrated module testing, thereby reducing the problems inherent in the design, integration, debugging and performance tests of such complex systems. Thus it is clear that the modularity principle also minimises the compatibility related problems inherent in the integration of complex systems.

Another important observation made during the testing of the system is that, because all the module functions and their inputs and outputs are known, online self-diagnostics facilities becomes an easy task to achieve. It is as simple as adding a separate module and watching the inputs and outputs from various modules and comparing them with the look-up table built using the experimental data.

The experimental results described in the paper not only reveal the benefits of the described control architecture and the used conceptual supervisory control system design approach, but also identify additional benefits such as possible future extension for online self-diagnostics capabilities for systems designed using the proposed design process. Three of the main achievements of the designed systems are the novel control system, its design approach, and the operational modes that extend the usefulness and functionality of the robot. The designed control system shows distinctive features that set it apart from many others discussed in the paper. This is mainly due to its composite structure that allows integration of various subsystems with complementary features, thereby making the overall control system unique and successful.

References

Ai, H., Zhang, B., and He, K.: 1994, Behaviour-based navigation for mobile robot, in: *Proc. of the 2nd Asian Conf. on Robotics and Its Applications*, pp. 565–568.

Abdelhameed, M. M.: 1999, Adaptive neural network based controller for robots, *Mechatronics* **9**(2), 147–162.

Albus, J., McCain, H., and Lumina, R.: 1987, NASA/NSB standard reference model for telerobot control systems architecture (NASREM), NBS Technical Report, No. 1235, NBS, Gaithersburg, MD, USA.

Arkin, R. C.: 1990, Integrating behavioural, Perceptual and world knowledge in reactive navigation, *Robotics Autonom. Systems* **6**, 105–122.

Arkin, R. C.: 1998, *Behaviour-Based Robotics*, MIT Press, Cambridge, MA.

Aylett, R. S., Coddington, A., Barnes, D. P., and Gray, J. O.: 1995, Planning and behaviours – A hybrid architecture for mobile robots, in: *IFAC'95 – Intelligent Autonomous Vehicles*, Espoo, Finland, pp. 85–92.

Beaufrere, B. and Zeghloul, S.: 1995, A mobile robot navigation using a fuzzy logic approach, *Robotica* **13**, 437–448.

Brooks, R.: 1984, Aspects of mobile robot visual map making, in: *Proc. of 2nd Internat. Symposium of Robotics Research*, pp. 369–375.

Brooks, R. A.: 1986, Robust layered control system for a mobile robot, *IEEE J. Robotics Automat.* **2**, 14–23.

Brooks, R. A. and Connel, J. H.: 1986, Asynchronous distributed control system for a mobile robot, in: *Proc. of SPIE, The International Society for Optical Engineering*, Vol. 727, pp. 77–84.

Brooks, R. A. and Flynn, A. M.: 1989, Fast, cheap and out of control, Technical Report, MIT AI Memo, No. 1227, MIT.

Brooks, R. A.: 1991a, Intelligence without reason, in: *Proc. of the 1991 Internat. Joint Conf. on Artificial Intelligence*, pp. 569–959.

Brooks, R. A.: 1991b, The role of learning in autonomous robots, in: *Proc. of the 4th Annual Workshop on Computational Learning and Theory*, pp. 5–10.

Chatila, R.: 1994, Control architectures for autonomous mobile robots, in: *Proc. of "From Perception to Action" Conference*, pp. 254–265.

Cho, S. B. and Song G. B.: 2000, Evolving CAM-brain to control a mobile robot, *Appl. Math. Comput.* **111**(2), 147–162.

Choi, W. and Latombe, J. C.: 1991, A reactive architecture for planning and executing robot motions with incomplete knowledge, in: *IEEE/RJS Internat. Workshop on Intelligent Robots and Systems*, pp. 24–29.

Dam, J. W. M. and Krose, B. J. A.: 1993, A neural-network that transforms occupancy grids by parallel Monte Carlo estimation, in: H. A. Wijshoff (ed.), *Computing Science in The Netherlands*, Utrecht, The Netherlands, pp. 121–131.

Dam, J. W. M. van, Kröse, B. J. A., and Groen, F. C. A.: 1996, Neural-network applications in sensor fusion for an autonomous mobile robot, in: L. Dorst, M. van Lambalgen, and F. Voorbraak (eds), *Reasoning with Uncertainty in Robotics*, pp. 263–277.

Dieguez, R. A., Raimundez, C., Sanz, R., Lopez, J., and Delgado, E.: 1995, An intelligent supervisory model for path planning and guidance of mobile robots in non-structured environments, in: *IFAC'95 – Intelligent Autonomous Vehicles*, Espoo, Finland, pp. 79–84.

Dorigo, M. and Schnepf, U.: 1993, Genetic-based machine learning and behaviour based robotics: A new synthesis, *IEEE Trans. Systems Man Cybernet.* **23**(1), 141–154.

Ebner, M. and Zell, A.: 2000, Centering behavior with a mobile robot using monocular foveated vision, *Robotics Autonom. Systems* **32**(4), 207–218.

Elfes, A.: 1987, Sonar-based real world mapping and navigation, *IEEE J. Robotics Automat.* **3**(3), 249–265.

Flynn, A. and Brooks, R. A.: 1989, Building robots: Expectations and experiences, in: *IEEE/RSJ Internat. Workshop on Intelligent Robots and Systems*, pp. 236–243.

Freyberger, F., Kampmann, P., and Schmidt, K.: 1990, Constructing maps for indoor navigation of a mobile robot by using an active 3D range imaging device, in: *IEEE/RSJ Internat. Workshop on Intelligent Robots and Systems*, pp. 143–148.

Gasos, J., Alegre, M. C. G., and Garcia, R.: 1992, Fuzzy strategies for the navigation of autonomous mobile robots, in: *Fuzzy Engineering Towards Human Friendly Systems*, pp. 1024–1034.

Gat, E.: 1992, Integrating planning and reacting in a homogenous asynchronous architecture for controlling real-world mobile robots, in: *Proc. of AAAI-92*, pp. 809–815.

Guglielmelli, E., Laschi, C., Teti, G., Fortanelli, R., and Dario, P.: 1997, A modular and distributed supervisory system for a semi-autonomous personal robot for household applications, in: *Proc. of ICAR'97 Internat. Conf. on Advanced Robotics*, Monterey, CA, USA, pp. 45–50.

Hall, E. L., Oh, S. J., and Kattan, E. U.: 1986, Experience with a robot lawn mower, in: *Conf. Proc. of Robots 10*, Chicago, USA, pp. 4.1–4.27.

Harris, C. J. and Moore, C. G.: 1989, Intelligent identification for autonomous guided vehicle using adaptive fuzzy-based algorithms, *Engrg. Appl. Artificial Intelligence* **2**, 267–285.

Heikkila, T. and Roning, J.: 1992, PEM-modelling: A framework for designing intelligent robot control, *J. Robotics Mechatronics* **4**(5), 437–444.

Holenstein, A. A. and Badreddin, E.: 1991, Collision avoidance in a behaviour based mobile robot design, in: *Proc. of the 1991 IEEE Internat. Conf. on Robotics and Automation*, Vol. 1, pp. 898–903.

Kasper, M., Fricke, G., Steuernagel, K., and Puttkamer, E. V.: 2001, A behaviour-based mobile robot architecture for learning from demonstration, *Robotics Autonom. Systems* **34**, 153–164.

Kurz, A.: 1996, Constructing maps for mobile robot navigation based on ultrasonic range data, *IEEE Trans. Systems Man Cybernet.* **26**(2), 233–242.

Kweon, S. I., Kuno, Y., Watanabe, M., and Onoguchi, K.: 1993, Sonar-based behaviour for a behaviour-based mobile robot, *IEICE Trans. Inform. Systems* **76D**, 479–485.

Lee, E. T.: 1995, Applying fuzzy logic to robot navigation, *Kybernetes* **24**(6), 38–43.

Leonard, J. J. and Durrant-Whyte, H. F.: 1991, Simultaneous map building and localisation for an autonomous mobile robot, in: *IEEE/RJS Internat. Workshop on Intelligent Robots and Systems*, pp. 1442–1447.

Li, W.: 1994a, Fuzzy logic based reactive behaviour control of an autonomous mobile system in unknown environments, *Engrg. Appl. Artificial Intelligence* **7**(5), 521–531.

Li, W.: 1994b, Fuzzy logic based perception action behaviour control of a mobile robot in uncertain environments, in: *Proc. of the 3rd IEEE Conf. on Fuzzy Systems*, Vol. 3, pp. 1626–1631.

Liscano, R., Fayek, R. E., and Tigli, J.: 1995, Using a blackboard to integrate multiple activities and achieve strategic reasoning for mobile-robot navigation, *IEEE Expert* **10**(2), 24–30.

Masoud, A. A., Masoud, S. A., and Boyoumi, M.: 1994, Robot navigation using a pressure generated mechanical stress field: The biharmonic potential approach, in: *Proc. of 1996 IEEE Internat. Conf. on Robotics and Automation*, Vol. 1, pp. 124–129.

Min, B. K., Cho, D. W., and Lee, S. J.: 1997, Sonar mapping of a mobile robot considering position uncertainty, *Robotics Comput. Integrated Manufacturing* **13**(1), 41–49.

Mountarlier, P. and Chatila, R.: 1993, Incremental free-space modelling from uncertain data by an autonomous mobile robot, in: *Geometric Reasoning for Perception and Action Workshop*, pp. 200–213.

Nehmzow, U. and McGonigle, B.: 1993, Robot navigation by light, in: *European Conf. on Artificial Life*, Brussels, Belgium.

Noborio, H., Kondo, K., and Noda, A.: 1990, A good extension method of free space in an uncertain 2D workspace by using an ultrasonic sensor, in: *IEEE/RSJ Internat. Workshop on Intelligent Robots and Systems*, pp. 665–671.

Novakovic, B., Scap, D., and Novakovic, D.: 2000, An analytic approach to fuzzy robot control synthesis, *Engrg. Appl. Artificial Intelligence* **13**(1), 71–83.

Palmer, R. J.: 1995, Guiding principles in controlling small automated vehicles, in: *Proc. of Power and Computing Conf., IEEE Wescanex 95 Communications*, pp. 366–370.

Reif, J. H. and Wang, H.: 1999, Social potential fields: A distributed behavioral control for autonomous robots, *Robotics Autonom. Systems* **27**(3), 171–194.

Rosenblatt, J. K: 1997, A distributed architecture for mobile navigation, PhD Thesis, The Robotics Institute, Carnegie Mellon University, Penssylvania, USA.

Santos, V., Goncalves, G. M., and Vaz, F.: 1994, Perception maps for the local navigation of a mobile robot: a neural-network approach, in: *Proc. of 1994 IEEE Internat. Conf. on Robotics and Automation* **3**, 2193–2198.

Sekiguchi, M., Nagata, S., and Asakawa, K.: 1989, Behaviour control for a mobile robot by dual hierarchical neural-network, in: *IEEE/RSJ Internat. Workshop on Intelligent Robots and Systems*, pp. 122–127.

Tieche, F., Facchinetti, C., and Hugli, H.: 1995, A behaviour-based architecture to control an autonomous mobile robot, in: *Proc. of the Internat. Conf. of Intelligent Autonomous Systems*, pp. 652–658.

Tubb, C. A. and Roberts, G. N.: 1998, Development of fuzzy behavioural controller for an autonomous vehicle, in: *UKACC Internat. Conf. on Control*, pp. 1717–1722.

Tunstel, E. and Jamshidi, M.: 1994, Fuzzy logic and behaviour control strategy for autonomous mobile robot mapping, in: *Proc. of 3rd IEEE Conf. on Fuzzy Systems*, Vol. 1, pp. 514–517.

Wei, L.: 1994, Fuzzy logic based perception action behaviour control of a mobile robot in uncertain environments, in: *Proc. of the IEEE Conf. on Fuzzy Systems*, Vol. 3, pp. 1626–1630.

Wu, S. F., Mei, J. S., and Niu, P. Y.: 2001, Path guidance and control of a guided wheeled mobile robot, *Control Engrg. Practice* **9**(1), 97–105.

Yavuz, H. and Bradshaw, A.: 1999, Conceptual design and development of a domestic robot, in: *Computer Aided Conceptual Design Workshop*, Lancaster University, Lancaster, UK, pp. 15–35.

Yavuz, H., Chandler, A., Bradshaw, A., and Seward, D.: 1999, Conceptual design and development of a navigation system for a mobile robot, in: *Computer Aided Conceptual Design Workshop*, Lancaster University, Lancaster, UK, pp. 65–85.

Yen, J., Pfluger, N., Lea, B., Murphy, M., and Jani, Y.: 1993, Employing fuzzy logic for navigation and control in an autonomous mobile system, in: *Proc. of the American Control Conf.*, San Francisco, USA, pp. 1850–1854.

Zelek, J. S. and Levine, M. D.: 1996, SPOTT: A mobile robot control architecture for known or partially known environments, Technical Report TR-CIM-96-01, Centre for Intelligent Machines, McGill University, Montreal, Canada.