Knowledge-based expert system using a set of rules to assist a tele-operated mobile robot

David Adrian Sanders[1], Alexander Gegov[2] and David Ndzi[3]

1. School of Engineering, University of Portsmouth

2. School of Computing, University of Portsmouth

david.sanders@port.ac.uk

Abstract — This paper firstly reviews five artificial intelligence tools that might be useful in assisting a tele-operator with driving a mobile robot: knowledge-based systems (including rule based systems and case-based reasoning), automatic knowledge acquisition, fuzzy logic, neural networks and genetic algorithms. Rule-based systems were selected to provide simple real time AI techniques to support tele-operated mobile robot operators with steering because they allow tele-operators to be included in the driving as much as possible and to reach their target destination, while providing assistance when needed to avoid an obstacle. The direction to a destination (via point) becomes an extra input along with an obstacle avoidance sensor system and the usual inputs from a joystick. A recommended direction is mixed with joystick position and angle. A rule-based system generates a recommended angle to turn the mobile robot and that is mixed with an input from a joystick in order to assist tele-operators with steering their mobile robots towards their destinations.

1. Introduction

Five artificial intelligence tools are reviewed: knowledge-based systems (including rule based systems and case-based reasoning), fuzzy logic, automatic knowledge acquisition, neural networks and genetic algorithms. Each artificial intelligence tool is outlined and briefly reviewed. A Knowledge-based expert system using a set of rules is selected to help a tele-operator to drive a mobile robot.

Applications of these tools have become more widespread and more complex mobile robot applications may require greater use of hybrid tools that combine the strengths of two or more of the tools. The tools and methods have minimal computation complexity and can be implemented on single robots or systems with low-capability microcontrollers. The appropriate deployment of the new AI tools will contribute to the creation of more efficient and effective mobile robot and tele-operated systems.

A rule-based system that describes knowledge in terms of IF…THEN…ELSE is selected for the tele operated mobile robot application. The tele-operated mobile robot obtains knowledge about the surroundings from sensors while moving towards a more overall end point. Assistance is provided to help tele-operators to avoid obstructions.

Systems presented in this paper help tele-operators drive when they cannot see the mobile robot or the environment (perhaps due to smoke) or when the robot is in a location away from the tele-operator.

Tele-operated systems are often open-loop. Operators communicate their desired direction and speed using a joystick. The robot then tends to move in the desired speed and direction. Tele-operator demands are processed and blended with inputs from the ultrasonics along with a more global destination end point to assist a tele-operator in driving their robot. Local and global planning are mixed inside a knowledge-based expert system using a set of rules to assist a tele-operated mobile robot. Local information from the ultrasonic sensor system[1] is blended with a global path.

Navigation for tele-operated mobile robots has been discussed within the literature [1-4]. Usually they have used a local algorithm and aimed to help tele-operators avoid obstacles[5] and suggest movements based on local sensors[4].

Some work has planned initial paths for mobile robots and then modified them locally [1] but that has rarely been to help a tele-operator. In this work, a local planner produces drive to motors attached to the driving wheels depending on input received from: the on-board ultrasonic sensors, the joystick, and the more globally defined targets. The tele-operated mobile robot responds rapidly to the human tele-operator and to changes in the environment ahead of the robot to avoid unexpected obstacles but tends to move towards a target destination whenever possible.

Huq *et al*. defined a fuzzy context-dependent blending of schemas [6] that eliminated a few of the restrictions of previous methods. Instead it used goal oriented navigation while avoiding obstacles within the robot path. Genetic algorithms have been blended with fuzzy logic to solve local mapping and position problems in [7]. That method automatically looked for a suitable plan to provide local environmental data. Bennewitz and Burgard described a randomized planning method to create real time routes within undefined environments without using vision [1], [8], while tracking a trajectory [9]. Hwang and Chang described avoidance techniques for car-like systems that used a fuzzy decentralized sliding-mode of control [10]. The potential field method was improved by Song and Chen by resolving some of the local minima problems [5] and Nguyen *et al* described obstacle avoidance using Bayesian Neural Networks [11].

A technique that improves a minimum-cost route is presented in this paper. A joystick mainly controls the speed but some simple AI systems also provides input [12-15]. The AI methods use perception based rules comparable to those described by Parhi & Singh, who used them for an autonomous mobile robot [1] and by Sanders *et al* [16] who considered a tele-operated mobile robot.

Algorithms trade path length against distance to an obstacle(s). Rules generate a suggested steering angle and that steering angle is merged with the input from a joystick to create the drive signals for the driving motors on the mobile robot.

The system and the techniques were successfully proven using simulation and then the sensors and microcontrollers were mounted on a Bobcat II mobile robot base (Fig. 1).

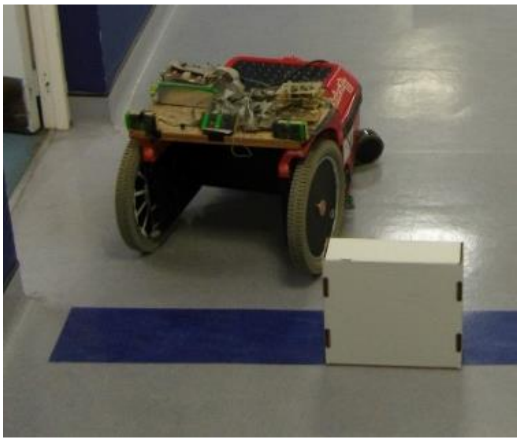

Fig. 1 Bobcat II mobile robot base avoiding an obstacle while being driven along a corridor

Many sensors can be used to avoid obstacles, for example: structured light or laser [17]; ultrasonics [18]; or infra-red [19]. The more global methods sometimes perform poorly indoors [20] but simpler and more local sensors can successfully determine position, for example: gyros, odometers, tilt, and ultrasonic [21][22]. Image processing can be useful when there is a clear view ahead of the camera but they can need more processing and they can be more complicated [23]. They are getting cheaper and computing power is quickly increasing [24]. The most accurate source of knowledge about the surroundings and situation comes from the human tele-operator but diminished visibility, separation and imperfect environmental information can reduce the ability of a human teleoperator [25].

Ultrasonics were selected for detecting ranges because it was inexpensive, uncomplicated, straightforward and rugged [26].

The paper continues with a review of the five artificial intelligence tools that were considered for this work followed by a description of the input from the sensors and joystick. Then the kinematics of the mobile robot base are described before discussing control and the artificial intelligence rule based tool selected. Then the testing and the results are described and the paper finishes with some discussion and conclusions.

2. Review of some artificial intelligence tools

Artificial Intelligence (AI) can improve teleoperation of mobile robots. AI has produced some useful tools for teleoperation that automatically solve problems normally requiring human brainpower. Five such tools are reviewed in this Section: fuzzy logic, knowledge-based systems, inductive learning, neural networks and genetic algorithms.

New advances are allowing seamless interactions between computers and people and the introduction of AI into teleoperation promises to make it more flexible, efficient and reliable. Tele operated mobile robots are exceeding human performance and as they merge with humans more intimately and we combine computer capacity with brain power to analyse, deliberate and make decisions, then we might be on the verge of a new assistive robot age.

A. Knowledge-based systems

Knowledge-based systems (sometimes called expert systems) are computer programs representing knowledge about solving problems. These systems typically have two principal parts, knowledge-bases and inference-mechanisms. Knowledge-bases hold knowledge about a domain that can be stated as arrangements of 'IF–THEN' rules, frames, factual statements, procedures, objects and cases.

Inference mechanisms manipulate stored knowledge to generate solutions. Knowledge manipulation methods include using constraints and inheritance (in object-oriented expert systems a frame-based expert systems), recovery and reworking of case examples (in a case-based system) and applying inference rules (within a rule-based system), corresponding to control procedures (forward or backward chaining) and search strategies (breadth or depth first).

Rule-Based Systems describe knowledge in terms of IF…THEN...ELSE. Decisions can be made using specific knowledge. They represent knowledge and decisions in ways that are understandable to human beings. Because of the rigid rule-base structure they can be poorer at handling uncertainty and imprecision. Typical rule-based systems have four fundamental components:

- the rules;
- an inference engine (or a semantic reasoner), that surmises information or acts depending on the interaction between the rules and the input(s);
- short-term memory;
- and user interfaces or alternative devices to input and output signals.

Case-Based Reasoning adapts solutions from earlier problems and applies them to existing problems. Solutions are stored in a database. The solutions can represent human experience. When a new problem is encountered, systems compare it with previous problems and selects a problem most similar to the new problem. It then acts using the previous solution and records whether the action was successful or a failure. Case-Based Reasoning is effective at representative knowledge in a way that is easy and well-defined for humans, but they can also learn from previous examples by creating extra new solutions.

Case-based reasoning has been formalized as a process with four steps:

i. Retrieve: Recover cases from short-term memory that are applicable to solving a target problem. Cases include a problem, its solution, and, often, comments concerning the way that a solution was originated.
ii. Reuse: Map a solution from a previous case onto the target problem. The solution may need to be adapted automatically to fit a new situation.
iii. Revise: After mapping a previous solution onto a target situation, test the solution and revise it if necessary.
iv. Retain: Once successfully adapted then store the resultant occurrence as a new case within short term memory.

CBR is frequently described as an expansion of Rule-Based Systems. Both CBR and Rule-Based Systems are useful for denoting knowledge clearly but CBR systems can also learn from the past by automatically creating new cases.

A lot of expert systems are created using ‘shells'; ready-made programs that are expert systems (including inferencing and knowledge storage but lacking domain knowledge). Sophisticated expert systems can be created using ‘development environments'. Development environments are more flexible than shells. They provide ways for operators to employ their own inferencing and ways of representing knowledge.

Expert systems are probably the most mature methods from amongst the five tools considered here and lots of development tools and commercial shells are available. The building of an system can be relatively simple once domain knowledge has been extracted, Because they are relatively easy to develop, a large number of applications have been created, for example for automatic robot programming and sequence planning.

B. Fuzzy logic

A rule-based expert system cannot handle a situation not explicitly included within their knowledge base (that is, situations not fitting within the ‘IF' statements within the rules). Rule-based systems cannot generate solutions when the encounter an unusual situation. They are consequently considered to be shallow systems which can fail in a ‘brittle' fashion, rather than gradually, as a human expert would.

Fuzzy logic reflects the qualitative and inexact nature of human reasoning. They can help an expert system to be more robust. Exact values for variables are exchanged for linguistic descriptions, represented by fuzzy sets. Based on this representation, the inferencing takes place. For example, an assembly speed of 35 thingamabobs per minute could be replaced by ‘normal' as a linguistic description of the variable ‘assembly speed. A fuzzy set defining the term ‘normal assembly speed ' might be:

normal assembly speed = 0.0/below 15 thingamabobs per minute +0.5/15−25 thingamabobs per minute +1.0/25−35 thingamabobs per minute +0.5/35−45 thingamabobs per minute +0.0/above 45 thingamabobs per minute.

The values 0.0, 0.5 and 1.0 are the degrees or grades of membership of the production ranges below 15 thingamabobs per minute (or above 45 thingamabobs per minute.), 15−25 thingamabobs per minute (35−45 thingamabobs per minute), and 25−35 thingamabobs per minute to the given fuzzy set. A grade of membership equal to 1 indicates full membership and a null grade of membership corresponds to total non-membership.

Knowledge within expert systems using fuzzy logic can be expressed as qualitative statements, (or fuzzy rules), such as ‘If apartment is at normal temperature, then set warmness inputs to normal'.

Reasoning procedures known as compositional rules of inference enable conclusions to be drawn by generalisation (interpolation or extrapolation) from qualitative information within a knowledge base. For example, when the normal assembly speed is perceived as ‘slightly below normal', a controlling fuzzy expert system may well determine that inputs should be increased to ‘slightly above normal'. Even though that conclusion may not have been covered by any fuzzy rule within the system.

Fuzzy Expert Systems use fuzzy logic to manage uncertainty produced by inadequate or partly corrupted data. Fuzzy logic uses a mathematical theory of fuzzy sets to mimic human logic. Humans easily deal with ambiguity when making decisions but computers still find it challenging.

Fuzzy logic has been used in mobile robotics, especially for control when domain knowledge has ben imprecise. Fuzzy Logic is useful when there is imprecision. For instance, for object recognition and scene interpretation. Fuzzy expert systems are suitable for ambiguous and imprecise situations. They cannot learn because system values cannot be changed.

C. Automatic knowledge acquisition

Learning programs often need a set of examples to use but it can be time consuming and difficult to get domain knowledge into a knowledge base. That can create a bottleneck during the construction of an expert system. Automatic knowledge acquisition techniques were created to deal with that.

An example of an approach is ‘divide-and-conquer'. Here attributes are selected according to a strategy that divides an example set into several subsets. A decision tree is then built to classify examples. The decision tree represents knowledge that is generalised from a set of specific examples. This can then be used to handle situations not covered by the example set.

Another example is a ‘covering approach'. An inductive learning program endeavours to locate groupings of attributes that are uniquely shared by examples within classes and then form rules with the IF part as combinations of those attributes and the THEN part as the classes.

Another example is the use of logic programming in place of propositional logic to depict examples and characterise new concepts. That uses a more potent predicate logic to characterise training examples and background knowledge and to convey new concepts. That allows results from induction to be defined as unspecific first-order clauses with variables.

There are many learning programs such as:

- ID3 (a divide-and-conquer program),
- FOIL (an ILP system adopting generalisation/specialisation methods),
- AQ program (which follows a covering approach),
- and GOLEM, (ILP system based on inverse resolution).

Most of these sorts of programs generate crisp decision rules but some algorithms have been created that also produce fuzzy rules.

Automatic learning has been tricky to use with tele-operated mobile robots because they require a set of examples in a rigid format and few mobile robot problems are described easily within rigid sets of examples. Automatic learning is generally more suitable for problems with discrete or symbolic attribute values rather than those with continuous-values. A recent application of inductive learning is in the control of a laser cutting robot.

D. Neural networks

Neural networks can capture domain knowledge from examples. However, they do not archive the acquired knowledge in an explicit form such as in rules or decision trees. They can readily handle both discrete and continuous data. They also have a generalisation capability (as for fuzzy systems).

Neural network models distribute computation between several simpler units called neurons. Neurons are interconnected and operate in parallel so that, neural networks can be called parallel-distributed-processing systems.

The most popular neural network is the multi-layer perceptron, which is a feedforward network: all signals flow in a single direction from the input to the output of the network. Feedforward networks can perform static mapping between an input space and an output space: the output at a given instant is a function only of the input at that instant. Recurrent networks, where the outputs of some neurons are fed back to the same neurons or to neurons in layers before them, are said to have a dynamic memory: the output of such networks at a given instant reflects the current input as well as previous inputs and outputs.

Implicit ‘knowledge' is built into a neural network during training. Some networks can be trained by presenting them with typical input patterns and the corresponding expected output patterns. Errors between the actual and expected outputs are used to modify weights on connections between neurons. This is “supervised training”. In a multi-layer perceptron, the back-propagation algorithm for supervised training is often adopted to propagate the error from the output neurons and compute the weight modifications for the neurons in the hidden layers.

Some neural networks are trained in an unsupervised mode, where only the input patterns are provided during training and the networks learn automatically to cluster them in groups with similar features.

Artificial Neural Networks typically have inputs and outputs, with processing within hidden layers in between. Inputs are independent variables and outputs are dependent. ANNs are flexible mathematical functions with configurable internal parameters. To accurately represent complicated relationships, these parameters are adjusted through a learning algorithm. Once trained then ANNs can accept new inputs and attempt to predict accurate outputs. To produce an output, the network simply performs function evaluation. The only assumption is that there exists some continuous functional relationship between input and output data. Like expert systems, they have found a wide spectrum of applications in almost all areas of robotics, addressing problems ranging from modelling, prediction, control, pattern recognition and optimisation.

E. Genetic algorithms

A genetic algorithm is a stochastic optimisation procedure inspired by natural evolution. A genetic algorithm can yield a global optimum solution within a complex multi-modal search space without specific knowledge about a problem.

Potential solutions to a problem must be represented as strings of numbers known as chromosomes and there must be a means of determining the goodness of each chromosome. A genetic algorithm operates on a group or population of chromosomes at a time, iteratively applying genetically based operators such as cross-over and mutation to produce fitter populations containing better solution chromosomes. The algorithm normally starts by creating an initial population of chromosomes using a random number generator. It then evaluates each chromosome. The goodness values of the chromosomes are used in the selection of chromosomes for subsequent operations. After the cross-over and mutation operations, a new population is obtained and the cycle is repeated with the evaluation of that population.

Genetic algorithms have found applications in tele-operation problems involving complex combinatorial or multi-parameter optimisation. Some recent examples of those applications are in Robot Path Planning.

F. Combining systems

The purpose of a hybrid system is to combine the desirable elements of different AI techniques within a single system. The different AI methods each have their own strengths and weaknesses. Some effort has been made in combining different methods to produce hybrid techniques with more strengths and fewer weaknesses. An example is a Neuro-Fuzzy system which seeks to combine the uncertainty handling of Fuzzy Systems with the learning strength of Artificial Neural Networks.

The nodes of a Fuzzy Network are fuzzy rule bases and the connections between nodes are interactions in the form of outputs from nodes that are fed as inputs to the same or other nodes. A fuzzy network is a hybrid tool combining fuzzy systems and neural networks due to its underlying grid structure with horizontal levels and vertical layers. This tool can be suitable for modelling the environment because separate areas can be described as modular fuzzy rule bases interacting in sequential / parallel fashion and feed forward / feedback context. The main advantages from the application of this hybrid modelling tool are better accuracy due to the single fuzzification-inference-defuzzification and higher transparency due to the modular approach used. These advantages can be crucial because of uncertainties in the data and the interconnected structure of the environment.

3. Selection of the rule based expert system

A knowledge-based expert system was selected to assist in teleoperation using an inference mechanism because they are relatively simple and good at representing knowledge and decisions in a way that is understandable to humans.

A knowledge base was created that contained domain knowledge as a combination of 'IF–THEN' rules. An inference mechanism manipulated the knowledge to produce solutions to driving problems.

The rule-based system used IF…THEN..ELSE to make decisions. It had four basic components: a list of rules, an inference engine, temporary working memory and a joystick user interface.

4. Mapping the environment ahead of the robot

The ultrasonics used were similar to those described in [27] and [28]. Ultrasonic sensors were mounted above the driving wheels on the front of the mobile robot. A measure of the distance to obstacles was provided by the time for a pulse to send and then be reflected back to a sensor. The mobile robot is described in more detail in [29].

An imaginary potential field was placed around objects within the range of the sensors [5][21]. The sensor system routinely adjusted the length of the pulses as ranges to objects changed. If no obstacles were sensed then the range-finder steadily lengthened the pulses in order to increase range until an object was found and that gave earlier warnings about potential problems.

False readings were filtered out using Histogramic In-Motion Mapping. Volumes in front of the mobile robot were divided into right and left lattices, with NEARBY, INTERMEDIATE and DISTANT, cells within the grid. A central volume was also created where the sensors overlapped, if objects were detected by both sensors. When something was detected ahead of the robot then it was categorised as NEARBY, INTERMEDIATE or DISTANT. The sensors were attached to the mobile robot chassis so that their rays over-lapped and enclosed the volume ahead of the robot.

If something was sensed then an element(s) associated with the cell was increased with a relatively large value, e.g.: ten, up to a limit of sixteen. Other cells in the grid reduced in value by a smaller amount, for example five, down to a final value of zero. This provided a histogrammic representation of the volume ahead of the robot. A cell quickly increased in value if an obstacle entered it. Random misreads in other cells only increased for a single misread and then the cell reduced in value. If the object appeared within another cell then that new cell quickly increased. When the object moved out of the original cell then it's value reduced to zero. Reliable ranges were attained in less than 0.5s.

5. Interpreting the joystick

A Penny & Giles joystick was used that contained two potentiometers. Joystick position was determined with two A/D converters.

The joystick data was in Cartesian coordinates but they were translated into polar coordinates: $J| \angle\theta$.

$|J|$ was a measure of how far a joystick was moved off-centre. That showed how fast an operator wanted the robot to go. Angle $\angle\theta$ was the preferred direction of travel.

The time that a joystick stayed in the same position suggested how confident the operator was in their decision.

$|J|$ was evaluated by means of:

$$|J| = \sqrt{((JSA*JSA)+(JSB*JSB))} \tag{1}$$

JSA and JSB were Cartesian co-ordinates.

$|J|$ and θ were used to establish the position of the joystick and therefor the desired direction and speed. Confidence and position were recorded in an array with each cell comprising 2 x values:

- "*Angle Confidence*" indicated whether the position of a joystick was remaining steady.
- "*Magnitude*" specified required mobile robot speed.

Joystick input was an input to the rule based system and it provided a confidence-level of user intentions.

The histogrammic depiction also represented a pseudo-integrator. If an operator held a joystick in one position, then the cell associated with that place increased in value. The other elements decremented. The element with the biggest value represented joystick position.

A computer procedure *JoystickArray* calculated which cell a joystick occupied and the associated "angle confidence" (*AngleConf*) increased. Other unoccupied cells decreased. So, histogram elements quickly reduced in value but built up in value slower.

JoystickArray cells increased to maximum in approximately 0.5s and reduced to zero in approximately 150 ms.

Weights to dictate the amount of increase or decrease were found experimentally. Specific weights could be set for individual human operators or for explicit tasks.

6. Kinematics of a BOBCAT II Base

The kinematics of the tele-operated mobile robot (Fig. 1) are described here. The robot had two large driving wheels at the front.

Movement and direction were accomplished by turning the driving wheels separately. Wheel radius was designated as r and diameter was then 2r. (Fig 2).

Using notation from [1], the driving wheels were W distance apart. C was the centre of gravity of the mobile robot. P was at the intersection of a line through the centre of the robot and another through the wheel axis. d was distance between C and P.

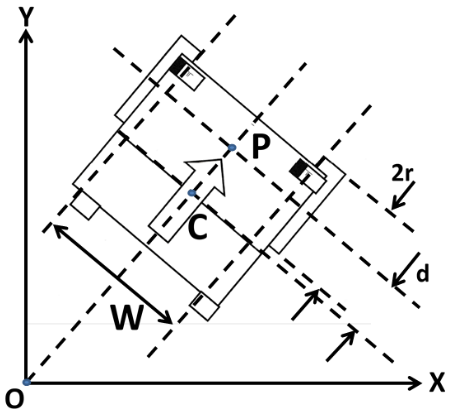

Fig. 2. Geometry of the tele-operated mobile robot

Kinematics for the tele-operated mobile robot is in Fig. 3.

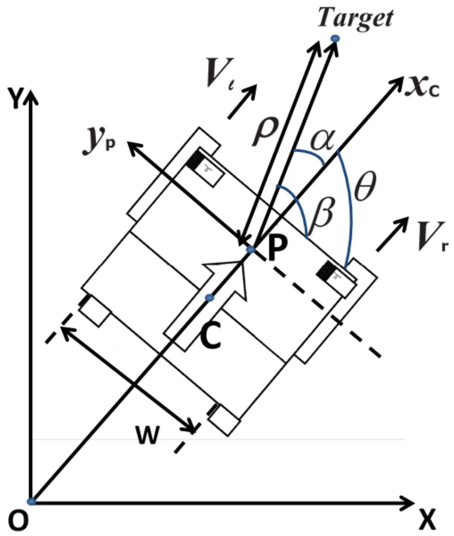

Fig. 3. Kinematics of the tele-operated mobile robot

It was assumed that no slip existed between the wheels and the floor.

$$v_{tang} = 1/2\ (v_{right} + v_{left}) \tag{2}$$

$$\omega_{tang} = 1/W\ (v_{right} - v_{left}) \tag{3}$$

$$v_{right} = r\omega_{right} \quad \text{and} \quad v_{left} = r\omega_{left} \tag{4}$$

where ω is angular velocity and v is linear velocity of the tele-operated mobile robot. The position of the mobile robot in global coordinates is [O X Y] and in vector notation is:

$$\boldsymbol{q} = [\ x_C\ y_P\ \theta]^T \tag{5}$$

where x_C and y_P are global coordinates of P (Fig. 2). θ is the orientation of $[P\ x_C\ y_P\ ]$, the local coordinate frame on the tele-operated mobile robot in Fig. 3. determined from the horizontal axis. The coordinates define the configuration of the mobile robot (5). The tele-operated mobile robot is rigid and wheels are assumed not to slip so that the tele-operated mobile robot can only move normal to the wheel axis. So, velocity at the point of contact with the ground (and orthogonal to the plane of the wheel) is zero.

$$(dy_P/dt) \cos \theta - (dx_C/dt) \sin \theta - d\theta/dt = 0 \tag{6}$$

Kinematics restrictions do not depend on time, and so are

$$A^T (\boldsymbol{q})\ d\boldsymbol{q}/dt = 0 \tag{7}$$

where $A(\boldsymbol{q})$ is an input matrix associated with constraints and

$$C^T A(\boldsymbol{q}) = 0 \tag{8}$$

where $C(\boldsymbol{q})$ is a full-rank matrix formed by a set of linearly independent vector fields covering the null space of $A^T(\boldsymbol{q})$. v_{tang} is a vector time function that can be found for times t from equations (7) and (8).

$$d\boldsymbol{q}/dt = C(\boldsymbol{q})\ \boldsymbol{v}_{tang} \tag{9}$$

For a tele-operated mobile robot the constraint matrix in (6) is

$$\mathbf{A}^T (\boldsymbol{q}) = [-\sin \theta\ \cos \theta\ -d] \tag{10}$$

and

$$\boldsymbol{v}_{tang} = [v\ \omega\ ]^T \tag{11}$$

Where ω is angular velocity and v is linear velocity of point P (along the tele-operated mobile robot axis). Therefore, the kinematics (9) can be described in a $d\boldsymbol{q}/dt$ matrix. As the tele-operated mobile robot only tends to move forwards then $v = -v_{,ang}$ and the system can be portrayed by a new simplified matrix. A controller was required to generate wheel velocities and steering angle was

Steering Angle = $(v_{left} - v_{right})/W$,

to drive the tele-operated mobile robot to follow the designated route.

7. Control and rules

v and ω were calculated to move the powered wheelchair from its current configuration, for example $\rho_0\, \alpha_0\, \beta_0$, to the target position.

Considering linear control[30]

$$v = K_\rho \rho \tag{12}$$

$$\omega = K_\alpha \alpha + K_\beta \beta \tag{13}$$

This closed-loop system could be depicted by a matrix to drive the mobile robot to $(\rho, \alpha, \beta) = (0,0,0)$, the target destination.

The controller was tested in simulation and then mounted onto the mobile robot. It had an overdamped response. Inputs from the joystick and ultrasonics were combined by means of a set of rules intended to avoid obstacles. Initial rules combined four inputs to avoid objects (fig 4). They were:

- Joystick steering angle;
- Distance to objects detected by both sensors;
- Distance to objects to the left of the robot;
- Distance to objects to the left of the robot.

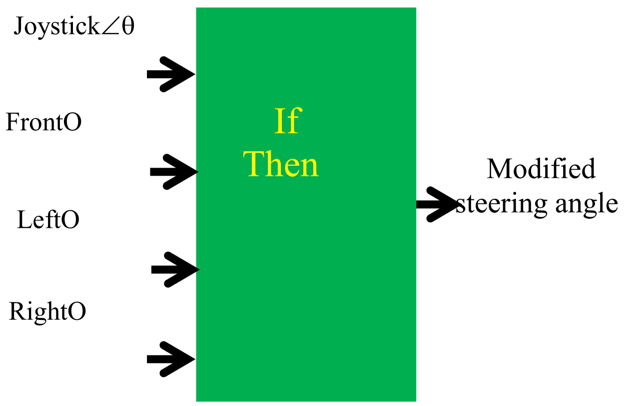

Fig. 4. Initial rule-based system.

Sensor input concerning the surroundings of the robot were used to modify the steering angle used in the controller. The suggested path was safe and efficient. If ∠θ was to the right of the tele-operated mobile robot then it tended to turn clockwise but if ∠θ was to the left then the tele-operated mobile robot turned anticlockwise.

The control systems worked well but in an effort to improve function if human sensors were impaired (for example, if an operator could not see the mobile robot for any reason), rules were modified to incorporate a new via point as a target destination to aid the tele-operators if they needed more help (fig. 5.). The rule based system now had a target via point to consider in addition to knowledge of the environment in front of the robot and a joystick steering angle. That increased the number of rules considerably.

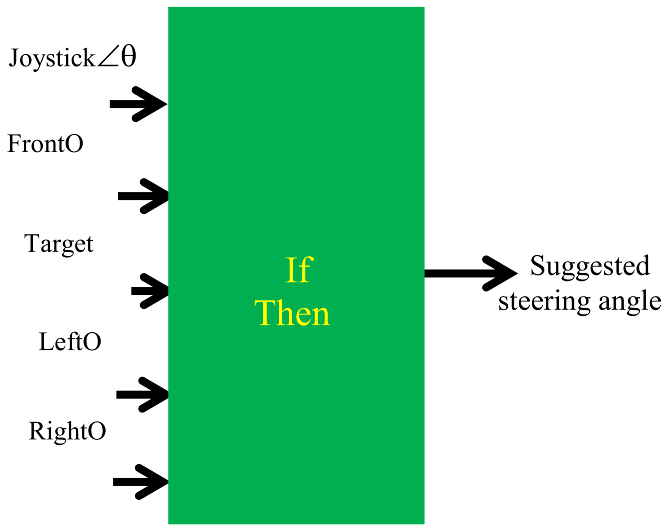

Fig. 5. Revised –rule-based system.

The rules in their revised from are described here:

CASE 1 - the obstacle and destination are on left of the tele-operated mobile robot:

Rule1: *If* Joystick=0^{o} *and* LeftO=INTERMEDIATE *and* RightO ≤ DISTANT *and* FrontO≤DISTANT *and* TargetAngle=75^{o}, *then* suggested change in steering angle=0^{o}

Rule2: *If* Joystick=0^{o} *and* LeftO=INTERMEDIATE *and* RightO ≤ DISTANT *and* FrontO≤DISTANT *and* TargetAngle =60^{o}, *then* suggested change in steering angle=-10^{o}

Rule3: *If* Joystick=0^{o} *and* LeftO=INTERMEDIATE *and* RightO ≤ DISTANT *and* FrontO ≤ DISTANT *and* TargetAngle =50^{o}, *then* suggested change in steering angle=-25^{o}

CASE 2 - the obstacle and destination are on the right of the tele-operated mobile robot:

Rule4: *If* Joystick=0^o *and* LeftO ≤ DISTANT *and* RightO = INTERMEDIATE *and* FrontO ≤ DISTANT *and* TargetAngle=75^o*, then* suggested change in steering angle=15^o

Rule5: If Joystick=0^o *and* LeftO= ≤ DISTANT and RightO = INTERMEDIATE *and* FrontO ≤ DISTANT *and* TargetAngle =60^o, *then* suggested change in steering angle=30^o

Rule6: *If* Joystick=0^o *and* LeftO= ≤ DISTANT *and* RightO = INTERMEDIATE *and* FrontO≤DISTANT *and* TargetAngle =30^o, *then* suggested change in steering angle=25^o

CASE 3 - an obstacle is in front and the destination is on the right:

Rule5: *If* Joystick=0^o *and* LeftO= NEARBY *and* RightO = NEARBY *and* FrontO ≤DISTANT *and* TargetAngle =20^o, *then* suggested change in steering angle=15^o

Rule6: *If* Joystick=0^o *and* LeftO= NEARBY *and* RightO = NEARBY *and* FrontO ≤DISTANT *and* TargetAngle =25^o, *then* suggested change in steering angle=20 o

Rule7: *If* Joystick=0^o *and* LeftO= NEARBY *and* RightO = NEARBY *and* FrontO ≤DISTANT *and* TargetAngle =30^0, *then* suggested change in steering angle=25^o

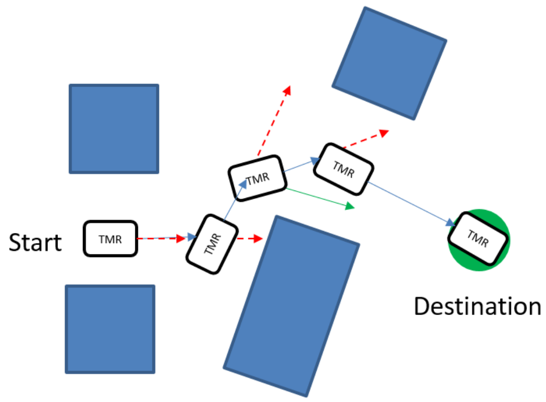

Fig. 6. Mobile robot moving through obstacles using the revised rule set showing approach directions (solid line) and calculated directions (dashed line).

The system worked better with the new rules and especially assisted drivers when human sensors were not working fully.

The path of the robot is shown again in Fig. 6. with the additional rules. The extra arrow is the angle to the destination.

8. Testing and results

A typical simulation of the system is shown in Fig. 7.

After the algorithms had successfully been tested in simulation, then the hardware and software were mounted onto the mobile robot base. A standard course at the University of Portsmouth was used for each test.

The tele-operated mobile robot avoided obstacles. When ultrasonic sensors detected an object close to the mobile robot, the robot avoided collision by turning away. That avoidance could be overruled by the joystick if the tele-operator wanted the robot to move close to the object.

Avoidance activated when sensors were DISTANT or closer. If sensors detected an object ahead while moving in the direction of the destination, then the mobile robot turned to move alongside the object. When there were no objects in the way, and the joystick was held in a forward position, the robot steered towards the target destination. That tended to reduce the time taken to get to destinations by a significant amount when vision was impaired (perhaps due to smoke etc). The rule-based system adjusted direction and quickly moved towards the target destination.

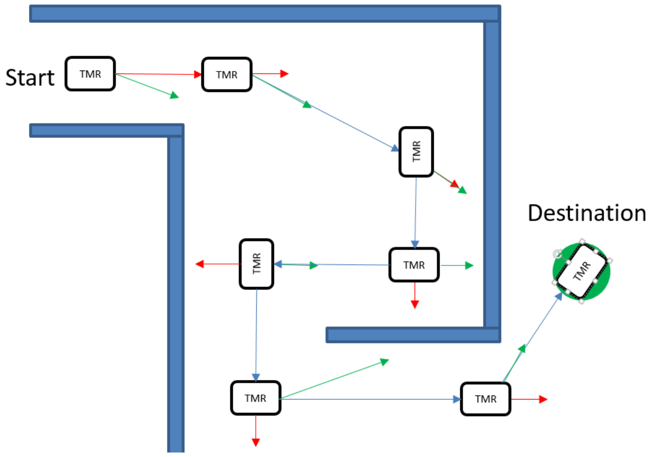

Fig. 7. Typical simulation using the revised set of rules showing the mobile robot avoiding local minima (for example the inner wall corners).

Results from simulation and from a real time experiment with the tele-operated mobile robot are shown in Fig. 6. and Fig. 8. As examples of how the systems were validated.

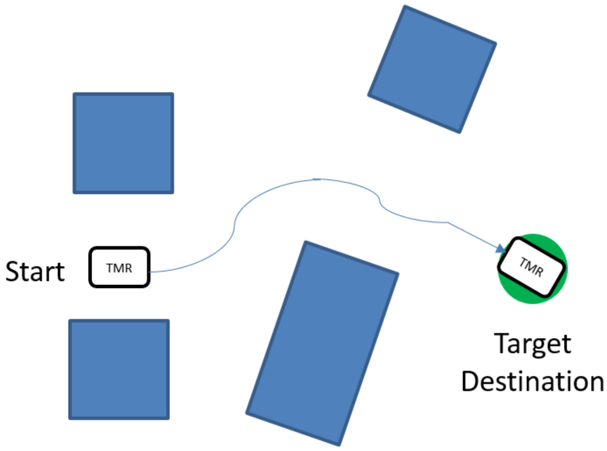

Fig. 8. Results from a real time experiment with the same rules applied.

Results were compared with those obtained using the systems in[1]. The rule-based system tended to perform better than the previous systems in terms of time taken to complete a path. Figure 9 shows a comparison of time taken by the systems as the tele-operated mobile robot was driven through a set of standard test environments at the University of Portsmouth.

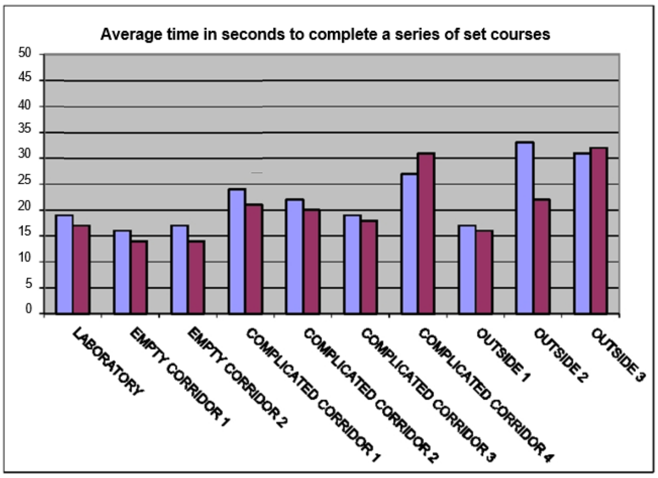

Fig. 9. Comparison between systems; showing average time taken to complete a series of set courses from a start point to a destination. Left hand bars show time taken without sensors to assist and right hand bars show time taken with sensors to assist.

Average time to complete a course was less for the new systems in most cases. There are two anomalies in figure 10. As environments became more complex they needed more turnings, for example to reach a target destination, the mobile robot may need to move through more than one room.

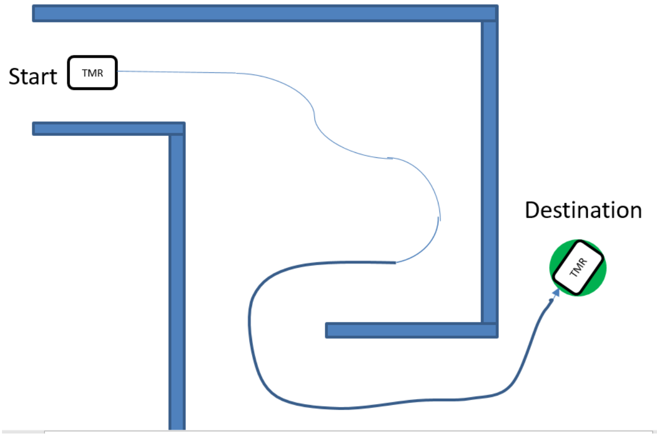

Fig. 10. Path of a robot using the revised rules when the tele-operator cannot see the robot.

Including a destination as an extra input made tele-operation a little less efficient in easy sections of a route and when the tele-operator could see what was happening. In those cases, the operator did not need a sensor system to help them. As an example, two routes are shown in Fig. 11 and Fig. 12.

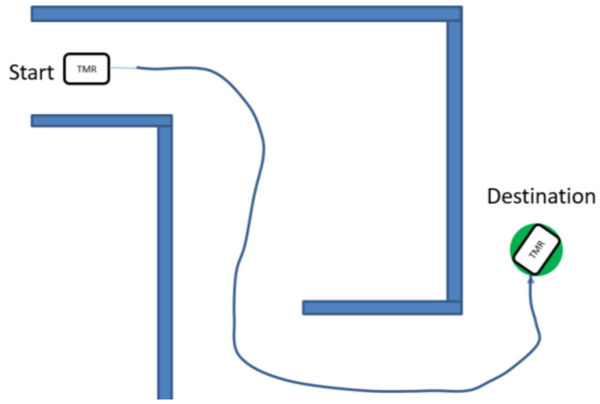

Fig. 11. Path of a robot using the revised rules when the tele-operator can see the robot.

In Fig. 10, the rules tended to pull the robot towards the destination. The original rules just depended on angular input from the joystick, $\angle\theta$. The path is less efficient but is completed despite not being able to see the robot. The difference is shown in Fig. 12.

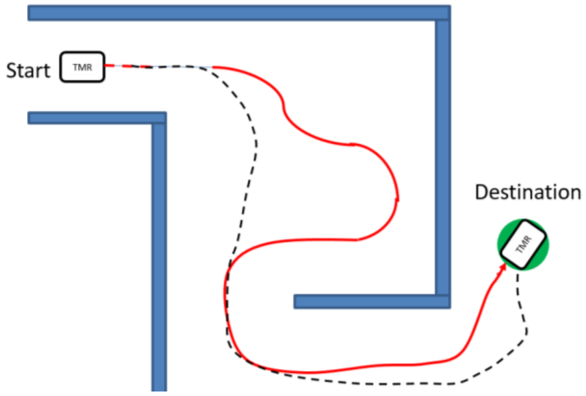

Fig. 12. Difference in the paths of a robot using the revised rules when the tele-operator can see the robot (dotted) and when the robot relies only on the sensors (solid line).

If a driver is capable of steering a robot quite well and they can see the robot then they can overcome the rules that might make the route less efficient.

The tele-operated mobile robots were able to reach destinations efficiently.

The methods provided a faster response in most cases and reduced the amount of computation time compared with other approaches and the rule-based system performed as effectively.

A real time path is shown in Fig. 11.

The tele-operated mobile robot needed to avoid static and moving obstacles and objects (for example human beings walking close to the robot).

When sensors received information about objects close to the tele-operated mobile robot, then the mobile robot avoided collision by turning away.

Collision avoidance was a high priority for the tele-operated mobile robot and initially overrode other behaviours, however if the joystick remained fixed (roughly) in a particular position then that input was integrated over time and the wishes if the tele-operator overrode that behaviour.

When the inputs from the sensors rose above a threshold within an array cell then avoidance was activated.

When the tele-operated mobile robot detected an obstacle in front while moving toward a target destination (via point) then wall-following behaviour was applied; the mobile robot tended to rotate to align with and then move parallel to the wall.

When sensors were not detecting anything, then the system drove in a direction that was an average between the angle to the target destination and the angle requested by the joystick. If the joystick was roughly in alignment with the direction of the target destination, then rules adjusted the direction of the mobile robot and sent it towards the target destination.

Results were compared with those obtained from recent alternative systems and the rule-based system performed well.

9. Discussion

Artificial intelligence has produced a number of powerful tools. This paper has reviewed some of those tools: knowledge-based systems, fuzzy logic, automatic learning, neural networks, ambient intelligence and genetic algorithms

The rule-base system selected was less good at handling uncertainty and is poor at handling imprecision because of the rigid structure. Case-Based Reasoning systems are often considered to be an extension of Rule-Based Systems. They are good at representing knowledge in a way that is clear to humans, but they also have the ability to learn from past examples by generating additional new cases.

Case-Based Reasoning could have been used because that can adapt solutions from previous problems to current problems. Solutions could be stored within a database. When a problem occurred that a system had not experienced, it could compare with previous cases and select one that was closest to the current problem. It could then update the database depending upon the outcome.

Without statistically relevant data for backing and implicit generalization, there is no guarantee that any generalization would be correct. However, all inductive reasoning where data is scarce is inherently based on anecdotal evidence.

The use of AI brings us to a point in history when our human biology can appear too slow and over-complicated. To overcome this, we are beginning to mix sensor systems and some powerful new technologies to overcome those weaknesses, and the longer we use that technology, the more we are getting out of it. We use less energy, space, and time, but get more and more assembly output for less cost.

The AI exceeded human performance in several tasks. As computers merge with us more intimately and we combine our brain power with computer capacity, then teleoperation should become easier and more efficient.

AI can reduce mistakes and increase efficiency. Time taken therefore reduces.

10.Conclusions

Applications of the AI tools discussed in this paper have become more wide-spread due to the power and affordability of present-day computers. Many new mobile robot applications may emerge and greater use may be made of hybrid tools that combine the strengths of two or more of the tools reviewed here. The tools have minimal computation complexity and can be implemented on single robots or systems with low-capability microcontrollers.

The rule-based system is safe and robust. It was uncomplicated and efficient in assisting a tele-operator with steering / driving a mobile robot. Rule based methods were applied successfully. The mobile robot rapidly acknowledged objects around it and assisted operators in completing their tasks.

Simulated paths were compared with tests in the laboratory and that validated the rules, the use of the rules and the robot systems. The system compared favourably with recent systems in the literature and that also validated the techniques.

Ongoing work is investigating the mixing of different AI tools in an effort to use the best of each technology.

References

[1] D.R. Parhi, and M.K. Singh “”Rule-based hybrid neural network for navigation of a mobile robot,” Proc. IMechE Part B: J. Engineering Manufacture, Vol. 224, 2009, pp 11103-1117.

[2] A.V. Nguyen, L.B. Nguyen, S. Su, and H.T. Nguyen “Shared Control Strategies for Human - Machine Interface in an Intelligent Robot,” 35th Annual Int Conf of IEEE-Engineering-in-Medicine-and-Biology-Society (EMBC), Osaka, JAPAN. Book Series: IEEE Engineering in Medicine and Biology Society Conference Proceedings, 2013, pp: 3638-3641.

[3] Z.R. Parhi, S.K. Pradhan, A.K. Panda, and R.K. Behra, “The stable and precise motion control for multiple mobile robots,” Appl. Soft Comput vol 9(2), 2009, 477- 487.

[4] D. A. Sanders. D. Ndzi, S. Chester and M. Malik “Adjustment of tele-operator learning when provided with different levels of sensor support while driving mobile robots”. Proc’ SAI Intelligent Systems Conference 2016, London, UK . *In press*.

[5] K. T. Song, and C. C. Chen, “Application of asymmetric mapping for mobile robot navigation using ultrasonic sensors,” J. Intell. Robotic Syst., 1996, 17(3), 243-264.

[6] R. Huq, G. K. I. Mann and R. G. Gosine, “Mobile robot navigation using motor schema and fuzzy context dependent behaviour modulation,” Appl. Soft Comput., 2008, 8(1), 422-436.

[7] M. Begum, G. K. I. Mann, and R. G. Gosine, “Integrated fuzzy logic and genetic algorithmic approach for simultaneous localization and mapping of mobile robots,” Appl. Soft Comput., 2008, 8(1), 150-165.

[8] M. Bennewitz and W. Burgard, “A probabilistic method for planning collision-free trajectories of multiple mobile robots,” In Proc 14th Euro Conf on AI, Berlin, Germany, 20-25 August 2000, pp. 9-15 (ECAI).

[9] W. Gueaieb and M. S. Miah, “An intelligent mobile robot navigation technique using RFID technology,” IEEE Trans. Instrum. Measmt, 2008, 57(9), 1908-1917.

[10] C. L. Hwang and N. W. Chang, “Fuzzy decentralized sliding-mode control of a car-like mobile robot in distributed sensor-network spaces,” IEEE Trans. Fuzzy Syst., 2008, 16(1), 97-109.

[11] A.V. Nguyen, and L.B. Nguyen, S. Su, H.T. Nguyen, “The Advancement of an Obstacle Avoidance Bayesian Neural Network for an Intelligent Robot”, 35th Int Conf of IEEE-Engineering-in-Medicine-and-Biology-Society (EMBC), Osaka, JAPAN. Book Series: IEEE Eng in Medicine and Biology Society Conf Proc, 2013, pp: 3642-3645.

[12] D. A. Sanders. “Using a self-reliance factor for a disabled driver to decide on the share of combined-control between a powered wheelchair and an ultrasonic sensor system”. IEEE Trans on Neural Systems and Rehabilitation Engineering. *In press*.

[13] D. Sanders, I. Stott, J. Graham-Jones, A. Gegov, and G.E. Tewkesbury, “Expert system to interpret hand tremor and provide joystick position signals for tele-operated mobile robots with ultrasonic sensor systems”. Ind Rob 38 (6). 2011, pp. 585-598.

[14] D.C. Robinson, D.A. Sanders, and E. Mazharsolook. “Ambient intelligence for optimal manufacturing and energy efficiency”. Assem. Autom. Vol 35 (3), 201, pp: 234-248

[15] D.A. Sanders, G. Tewkesbury, A. Gegov, *et al*. “Fast transformations to provide simple geometric models of moving objects”. Proc’ Intelligent Robotics & Application Confs, Pt I Vol 9244, 2015, pp: 604-615.

[16] D. Sanders, H. Sanders, D Ndzi and A. Gegov, “Rule-based system to assist a tele-operator with driving a mobile robot” IEEE Proc of SAI Intelligent Systems Conf, 2016, pp: 987–994

[17] J. Larsson M., Broxvall and A. Saffiotti, "Laser-based corridor detection for reactive Navigation. Ind Rob: An int' jnl, 35 (1), 2008, pp 69-79.

[18] D. Sanders, J. Graham-Jones, and A. Gegov, "Improving ability of tele- operators to complete progressively more difficult mobile robot paths using simple expert systems and ultrasonic sensors," Ind Rob: An int' jnl, 37 (5). 2010, pp. 431-440.

[19] S. Lee "Use of infrared light reflecting landmarks for localization. Ind Rob: An int' jnl 36 (2), 2009, pp: 138-145.

[20] V. Milanes, J. Naranjo, C. Gonzalez *et al*, "Autonomous vehicle based in cooperative GPS and inertial systems", Robotica 26, 2008, pp 627-633.

[21] D. Sanders and I. Stott, "A new prototype intelligent mobility system to assist tele-operated mobile robot users", Ind Rob 26(6), 09, pp: 466-475.

[22] Y. C. Chang and Y. Yamamoto "On-line path planning strategy integrated with collision and dead-lock avoidance schemes for wheeled mobile robot in indoor environments," Ind Rob: An int' jnl, 35 (5), 2008, pp 421-434.

[23] D. Sanders, "Progress in machine intelligence", Ind Rob 35 (6), 2008, pp: 485-487.

[24] D. A. Sanders, G. Lambert, and L. Pevy, "Pre-locating corners in images in order to improve the extraction of Fourier descriptors and subsequent recognition of shipbuilding parts", Proc' Inst Mech Eng PART B-Jrnl of Engineering manufacture 223, vol 9, 2009, pp: 1217-1223.

[25] D. Sanders, "Comparing speed to complete progressively more difficult mobile robot paths between human tele-operators and humans with sensor-systems to assist," Assem. Autom. vol 29 (3), 2009, pp 230-248.

[26] D. Sanders, "Analysis of the effects of time delays on the tele-operation of a mobile robot in various modes of operation," Ind Rob vol 36 Issue: 6, 2009, pp: 570-584. 2.

[27] D. Sanders and A. Gegov. "Different levels of sensor support changes the learning behaviour of wheelchair drivers". Assistive Technology. *In press*.

[28] D. A. Sanders and N.Bausch, (Ed H. Liu). "Improving Steering of a Powered Wheelchair Using an Expert System to Interpret Hand Tremor". Proc of Intelligent Robotics and Applications (Icira 2015), Pt Ii Volume: 9245, pp: 460-471. 2015.

[29] D. Sanders, I. J. Stott and D. C. Robinsosn e*t al.* "Analysis of successes and failures with a tele-operated mobile robot in various modes of operation". Robotica 30, pp: 973-988. 2012.

[30] D. A. Sanders, G. E. Tewkesbury, and I. J. Stott, *et al*. "Simple expert systems to improve an ultrasonic sensor-system for a tele-operated mobile-robot". Sensor Review 31 (3), pp 246-260. 2011.

[31] D. Sanders and A. Geov. "AI tools for use in Assem. Autom. and some examples of recent applications. Assem. Autom. Vol 33 (2), pp: 184-194. 2013.

[32] D. A. Sanders, G. E. Tewkesbury, and D. Ndzi *et al*. "Improving automatic robotic welding in shipbuilding through the introduction of a corner-finding algorithm to help recognise shipbuilding parts". Journal of Marine Science and Technology Vol 17 (2), pp: 231-238. 2012.

[33] D. Sanders, G. Lambert, and J. Graham-Jones *et al*. "A robotic welding system using image processing techniques and a CAD model to provide information to a multi-intelligent decision module". Assem. Autom. Vol: 30 (4), pp: 323-332. 2010.

[34] D. A. Sanders, and G. E. Tewkesbury. "A pointer device for TFT display screens that determines position by detecting colours on the display using a colour sensor and an Artificial Neural Network". Displays Vol 30 (2), pp: 84-96. 2009.

[35] J. Bergasa-Suso, D. A. Sanders, and G. E. Tewkesbury. "Intelligent browser-based systems to assist Internet users". IEEE Transactions on Education Vol 48 (4), pp: 580-585. 2005.

[36] D. A. Sanders, Y. Tan, I Rogers *et al*. "An expert system for automatic design-for-assembly". Assem. Autom. 29 (4), pp: 378-388. 2009.

[37] D. A. Sanders, G. Lambert, and L. Pevy. "Pre-locating corners in images in order to improve the extraction of Fourier descriptors and subsequent recognition of shipbuilding

parts". Proc of IMechE Part B-Journal of Engineering Manufacture 223 (9), pp: 1217-1223. 2009.

[38] D. Sanders. "Controlling the direction of walkie type forklifts and pallet jacks on sloping ground". Assem. Autom. 28 (4), pp: 317-324. 2008.

[39] D. A. Sanders. "Progress in machine intelligence". Ind Rob: An int' jnl 35 (6), pp: 485-487. 2008.

[40] A. Geov, N. Gobalakrishnan and D. A. Sanders. "Rule base compression in fuzzy systems by filtration of non-monotonic rules". Journal of Intelligent & Fuzzy Systems Vol 27 (4), pp: 2029-2043. 2014.