New Hybrid Control Architecture for Intelligent Mobile Robot Navigation in a Manufacturing Environment

Najdan Vuković
PhD student
University of Belgrade
Faculty of Mechanical Engineering

Zoran Miljković
Associate Professor
University of Belgrade
Faculty of Mechanical Engineering

This paper presents a new hybrid control architecture for Intelligent Mobile Robot navigation based on implementation of Artificial Neural Networks for behavior generation. The architecture is founded on the use of Artificial Neural Networks for assemblage of fast reacting behaviors, obstacle detection and module for action selection based on environment classification. In contrast to standard formulation of robot behaviors, in proposed architecture there will be no explicit modeling of robot behaviors. Instead, the use of empirical data gathered in experimental process and Artificial Neural Networks should insure proper generation of particular behavior. In this way, the overall architectural response should be flexible and robust to failures, and consenquently provide reliableness in exploitation. These issues are important especially if one takes under consideration that this particular architecture is being developed for mobile robot operating in manufacturing environment as a component of Intelligent Manufacturing System.

Keywords: *robotic control architecture, behavior based robotics, intelligent mobile robot, artificial neural networks, navigation, intelligent manufacturing systems.*

1. INTRODUCTION

At the beginning of the 21st century manufacturing and closely related technologies are more than ever correlated to fast growing market requirements and intensively coupled with diverse customer demands [1]. The ever increasing production complexity and growing tendency for delivery time cutting of products, as well as the need for "make to order" rather than "make to stock" manufacturing [2,3], imposes the development and implementation of advanced paradigms and engineering solutions able to tackle with these sophisticated issues. New methods, fast growing research fields, design principles and newly developed and defined paradigms should result in improved manufacturing technology and provide desired quality of products and services [4].

The advanced manufacturing paradigms like Computer Integrated Manufacturing (CIM) and Intelligent Manufacturing Systems (IMS) may provide more integrated manufacturing environments developed on the basis of modern software architectures and information technologies [2,5,6].

On the other hand, the implementation of advanced concepts and paradigms of manufacturing in order to overcome problems that arise on the shop floor on daily level is a hard scientific and engineering challenge. The main problems facing the integration of novel concepts are inherent unpredictability and extensive complexity of manufacturing environment. This dynamic behavior imposes the analysis of each and every element of manufacturing processes, so that progress could be achieved.

One of these elements is a Material Handling (MH) which is defined by the Material Handling Industry of America as "the movement, storage, protection and control of materials throughout the manufacturing and distribution process including their consumption and disposal" [2]. As one may easily infer, material handling process must be performed safely, efficiently, at low cost and without damage to the goods. Although often being overlooked, material handling is an important issue in production since the cost of material handling is a significant portion of the total production cost averaging around 20 % of the total manufacturing labor cost [2]. Hence, it is easy to see why this hard problem should be in the focus of a research community, side by side with other quintessential problems in the manufacturing. However, the problem of material handling could not be solved directly since it covers a vast field of problems spreading from storage, protection, delivery planning, optimization issues, transport, etc. Therefore, these multidisciplinary research fields are to be analyzed individually.

As it has been stated, one of the elements of the MH concept is the transport of materials, goods and products. Transport issues in manufacturing are a part of Material Transport System (MTS). Conventional solutions of material transport are based on forklift trucks (industrial trucks), conveyer belts, Automatic Guided Vehicles (AGV), etc. Each of these solutions is fully integrated in manufacturing processes and successfully implemented in industry worldwide. However, due to the inherent complexity of the manufacturing environment, conventional solutions still do not solve all aspects of material transport problem.

Received: February 2009, Accepted: March 2009
Correspondence to: Najdan Vuković
Faculty of Mechanical Engineering,
Kraljice Marije 16, 11120 Belgrade 35, Serbia
E-mail: nvukovic@mas.bg.ac.rs

Namely, each conventional solution has been exclusively developed for a particular purpose, resulting in a specific implementation. On the other hand, the behavior of these systems should be constantly supervised and checked for the errors. Therefore, one may see that a human operator is an essential component and much needed factor for reliable implementation of these concepts. This limitation of conventional solutions results in increased overall expenses and cost of manufacturing. Consequently, the cost of products gets higher, which decreases the competitive potential of the company on the market.

Having previous analysis in mind, this paper is a part of the research that addresses the issue of Intelligent Mobile Robots (IMR) implementation in the realm of manufacturing and integration in the MTS [4,5]. This advanced application of the IMR should solve some aspects of the transport problem and provide guidelines for the research in the future.

However, it should be emphasized that this progressive integration of the IMR in manufacturing systems is still a challenge for the research community and the general framework has not been established. On one hand, the role of mobile robot in the manufacturing environment should be defined following the basic framework established through integration levels in the CIM (level of mechanical integration, level of communication integration and integration using knowledge) [4,5]. On the other hand, the aspects related to intelligent behavior of the IMR in the known/unknown environment are to be analyzed and introduced. Nevertheless, despite the overall complexity of the IMR integration in manufacturing processes, the novel and advanced manufacturing paradigms, like the CIM and particularly the IMS, point out the necessity to have installed intelligent systems capable to solve the majority of specific complex problems in manufacturing without human interfering, whatsoever. The main scientific and engineering challenge in the IMS research field is to develop fully autonomous factory which would not require human operators. This progressive idea brings us to The Black Factory concept [7], the factory where there are no lights, since there are no human operators.

This paper is organized as follows. The second section of the paper introduces the existing deliberative, reactive and hybrid control paradigms for the development of the IMR and the basic framework for their implementation. In the third section the newly proposed hybrid architecture is provided. Finally, at the ending of the paper, the discussion and conclusion are provided along with expected results that this new architecture should achieve.

2. PROBLEM FORMULATION

Mobile robots are being extensively used in various fields stretching from everyday human activities to their advanced implementation in diverse and hazardous environments. With each and every achieved success the research field gains more attention resulting in progressive frameworks and successful applications that previously could not have been imagined. However, each implementation of mobile robots implies particular concepts and engineering solutions able to deal with problems emerging on the daily level. Therefore, the concept of general framework is suited exclusively for the problems being analyzed.

As an intelligent agent, a mobile robot needs the ability to perceive information from the working environment by using a particular sensor or group of sensors and based on that information make a decision of its responses. In accordance with the state of the environment that decision should result in appropriate reaction of the mobile robot enabling it to successfully operate. Having this in mind, the basic characteristics of intelligent agents have to be defined. Any intelligent agent should be developed based on the following building blocks [4,8]: the sensing ability, the perceptive ability, the knowledge acquisition ability, the learning ability, the inference ability, the decision making ability and the acting ability. Each of these abilities provides an opportunity to generate desired and specific response of a mobile robot relative to the working environment.

However, the essential question is: How can these abilities be augmented in the various robotic systems? In other words, how can integration of these abilities be achieved? This question undoubtly leads us to the study of robotic architectures. Robotic architecture is a software system that provides languages and tools for the construction and development of robotic systems and in a broader sense, robotic architecture may be defined as *the discipline devoted to the design of highly specific and individual robots from a collection of common software building blocks* [9]. Therefore, robotic architecture defines the flow of information, how it is being perceived, how it is being transformed and finally, how decisions are being made. The distinction between definition of architecture in the computer science sense and robotic system sense could be easily noticed.

The basics of information gathering, understanding, representation and acting fall in the realm of Artificial Intelligence (AI). As it was defined in [10] the AI attempts to understand intelligent entities. Therefore, the AI provides scientific framework (mathematical, philosophical, gnoseological, etc.) for development of intelligent agents and consequently intelligent systems. As an agent operating in an known or unknown environment robotic system have to be intelligent agents and therefore their control architectures, which are essential for perceiving and understanding information, have to be developed on the AI basis.

2.1 Deliberative control

In the 1980's, the dominant view in the AI community was that the control system for an autonomous mobile robot should be decomposed into three functional elements: a sensing system, a planning system, and an execution system [11] (Fig. 1). The sensing system should provide information of the environment and the robot state as well, the planning system is to plan future robot actions, and finally, the execution system is to perform action selected by the planning system.

This approach has dominated ever since the AI as a distinct field had emerged. The deliberative paradigm heavily relies on representational knowledge and deliberative reasoning methods, and one may say that in those years this was a mainstream. The Sense-Plan-Act (SPA) approach has unidirectional flow of information (from sensors to world model to plan to effectors) and reverse flow is not achievable.

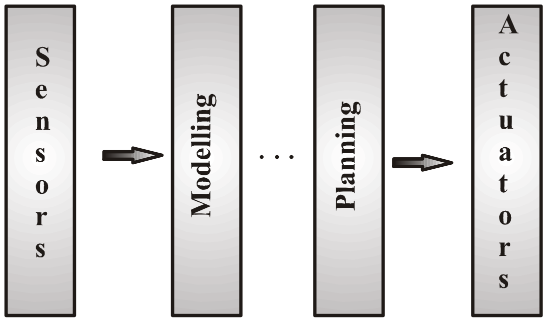

Figure 1. Information flow in hierarchical paradigm

The major drawback of deliberative paradigm (in the literature this paradigm is often referred to as traditional or conventional paradigm) is time and computational cost. Namely, in order to perform necessary calculations defined by the world model and the planning system the mobile robot hardware (in the computer science sense) had to have enormous memory and processing capabilities. The hierarchical paradigm suffers from difficulty of real time interaction with the working environment.

2.2 Behavior-based control

Being unable to achieve reliable implementation of conventional approach for each and every intelligent mobile robot being developed, the robotic community turned towards new solutions. These new concepts should have provided frameworks able to tackle problems that conventional approach had not been able to. The brand new paradigm emerged in the 1980's and it was named the Behavior Based Robotics (BBR). The overall architectural disposal is presented in Figure 2.

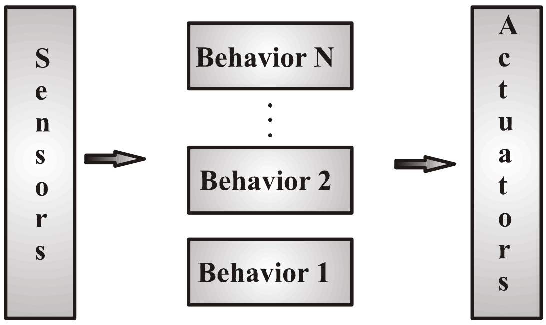

Figure 2. Information flow in reactive paradigm

The BBR architecture consists of several layers. The first one gathers information from the working environment. The second is founded on the particular form of transfer functions called behaviors, which transforms sensor input into predefined response. Finally, the last layer is to perform the action based on the output of the specific behavior. Two most recognizable BBR approaches will be briefly presented in the subsequent part of the paper.

The first proposal in the BBR style emerged in the mid 1980's. Namely, Prof. Rodney Brooks of MIT's Artificial Intelligence Laboratory proposed a new solution for mobile robot developing. In his seminal paper [12] Brooks argues that use of traditional hierarchical structure led researchers in the wrong direction of the main research goal, which is to build a mobile robot able to operate autonomously in the environment. Namely, the dominant focus on the planning system and explicit symbolic representational knowledge is timely and computationally costly, restraining robotics community to tackle those problems for the time being. In a way, Brooks eliminated the planning system from the control architecture and focused exclusively on the sensing and acting modules. He advocated the use of layered, but not hierarchical, architecture that could provide a mobile robot with autonomous capabilities. The low level layers are built for obstacle avoidance, while the higher layers for more abstract actions. The architecture was named *subsumption architecture* (Fig. 3).

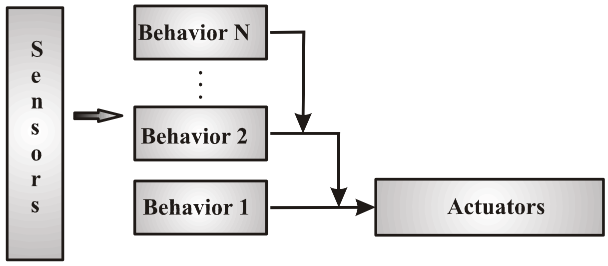

Figure 3. Subsumption architecture

In the subsumption architecture each behavior is realised in the form of Augmented Finite State Machine (AFSM) [12]. Coordination of behaviors is achieved by the priority-based arbitration via inhibition and suppression. Therefore, the overall output of architecture is being generated by the highest active behavioral module. On the other hand, although the subsumption architecture is the most radical of proposed reactive approaches, it should be pointed out that paradigm represents small step towards making traditional Sense-Plan-Act approach work.

The main morals of subsumption approach towards building mobile robots are vividly stated in the following messages: *the world is its best model* and *the planning is just another way of avoiding what to do next* [12]. Throughout years, a great number of subsumption based robots have been developed and put in the service [13], effectively proving the advantages of this reactive approach.

Approximately at the same period, Prof. Ronald Arkin from Georgia Institute of Technology proposed yet another reactive architecture based on perceptual schemas [9]. This approach is more than the subsumption architecture strongly motivated by the biological sciences. Namely, the motor schema theory explains motor behavior in terms of the simultaneous control of many different activities [9]. Each behavior produces the output in the vector form, while the overall response of the system is achieved by vector summation. It should be stressed that motor schemas represent more than a behavior, namely, by defining behaviors using this reactive paradigm, the control law is defined as well.

One of the most important issues in the BBR is coordination of behaviors. The main advantages of the BBR approach is the ability to incrementally build mobile robot, layer upon layer, starting from the lower levels (move, avoid obstacles, etc.) and finishing to the higher levels (discover new areas, modify the world, etc.). However, coordination among behaviors is a major issue. To be more specific, which behavior is to be triggered and set into action, if two or more behaviors are triggered which one should send commands to motors, etc. are main questions. The subsumption architecture and the motor schemas approach are founded on quite distinct mechanism for solving this issue. Namely, in the subsumption architecture behaviors are coordinated by means of suppression and inhibition. The higher levels are able to subsume (thus the name) the inputs as well as outputs of lower layers (Fig. 3). Unlike the subsumption architecture, relying on competitive selection of behaviors, the motor schemas advocate the use of cooperative coordination which provides an ability to simultaneously use the output of more than one behavior, capturing their particular influence in overall output (Fig. 4). However, the framework of behavior coordination heavily depends on the designer choice.

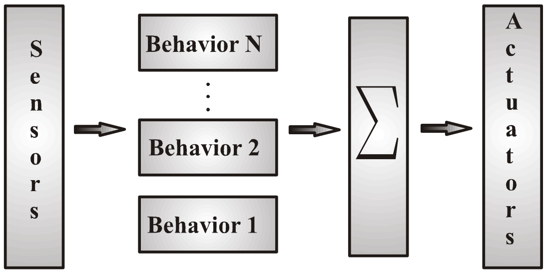

Figure 4. Motor schemas

2.3 Hybrid approach

Although the BBR establishes the framework for successful mobile robot design, it still does not solve some of the problems of mobile robot implementation in real world. It should be pointed that the BBR heavily relies on the following assumptions: *the environment lacks temporal consistency and stability; the robot's immediate sensing is adequate for the task at hand; it is difficult to localize a robot relative to a world model; symbolic representational world knowledge is of little or no value* [9]. To autonomously operate in real world, the mobile robot needs a variety of abilities that exceeds reactive paradigm hence, some aspects of deliberative planning are essential for real world implementation. In contrast to paradigms that have been introduced so far, the hybrid approach involves the benefits from both hierarchical and reactive paradigms. On the low level a BBR concept is applied. Whether it is a subsumption style or motor schemas is not of importance at this point. The role of reactive layer is to generate the action based on the environment's stimuli. On the other hand, the deliberative layer is to tackle with high level issues that are not solvable by means of reactive control. For instance, this layer should determine the position and orientation of a mobile robot relative to the environment or, if the goal is achieved, the deliberative layer should provide answers to the following question: Where should I go now?

The hybrid paradigm emerged on the verge of the 1990's and one of the first architectures was developed by Arkin itself [9]. The architecture's name is Autonomous Robot Architecture (AuRA). Three layered architecture (3T) is the second significant architecture, introduced by Erann Gat [11].

3. NEW HYBRID CONTROL ARCHITECTURE FOR MOBILE ROBOT NAVIGATION IN A MANUFACTURING ENVIRONMENT

Although robotics has achieved a number of great successes [14,15], the general framework for the development on architectural level has not been defined by the community. Researchers argue that the aforementioned general framework is hard to develop [10,12], since this problem is closely related to questions like: *What is intelligence? How can it be defined? Is it possible to achieve intelligent behavior in artificial agents?*, etc. These questions are part of the multidisciplinary study that should embody cognitive and neuro sciences, artificial intelligence, philosophy, etc., in order to tackle this sophisticated and in a way mystic issue. On the other hand, the research community argue [10,12,16] that it is too early (or even unnecessary generally speaking) to define general framework. Therefore, the raised questions are to be solved gradually and the development of robotic control architectures is one minor step towards general solution.

The development of control architecture heavily depends on the environment, the task robot is being designed to do, hardware components, available funding, etc. Hence, the robot designer is left with the freedom of choice. It could be easily verified that each and every success in the field [14,15] stands upon different kinds of control architectures. Therefore, one may conclude that working characteristics of the environment have severe influence and, in a way, "define" the architectural solution.

As it has been stated in the previous part of the paper, the manufacturing environment imposes specific abilities that a mobile robot, as an element of the MTS, ought to have. For instance, fast reacting behaviors for static or dynamic obstacle avoidance, perception and world representation ability – to enable information gathering and processing, map building ability – to insure the robot would be able to localize itself relative to the environment, inference and decision making ability – to provide the robot with ability to "understand" gathered information and based on that particular information make a reliable decisions, etc. Having this in mind, it is quite obvious what the prerequisites are required.

Based on previous analysis, the authors propose a new hybrid control architecture for mobile robots operating in a manufacturing environment. The architecture is founded on the basis of hybrid approach. The architecture can be seen in Figure 5.

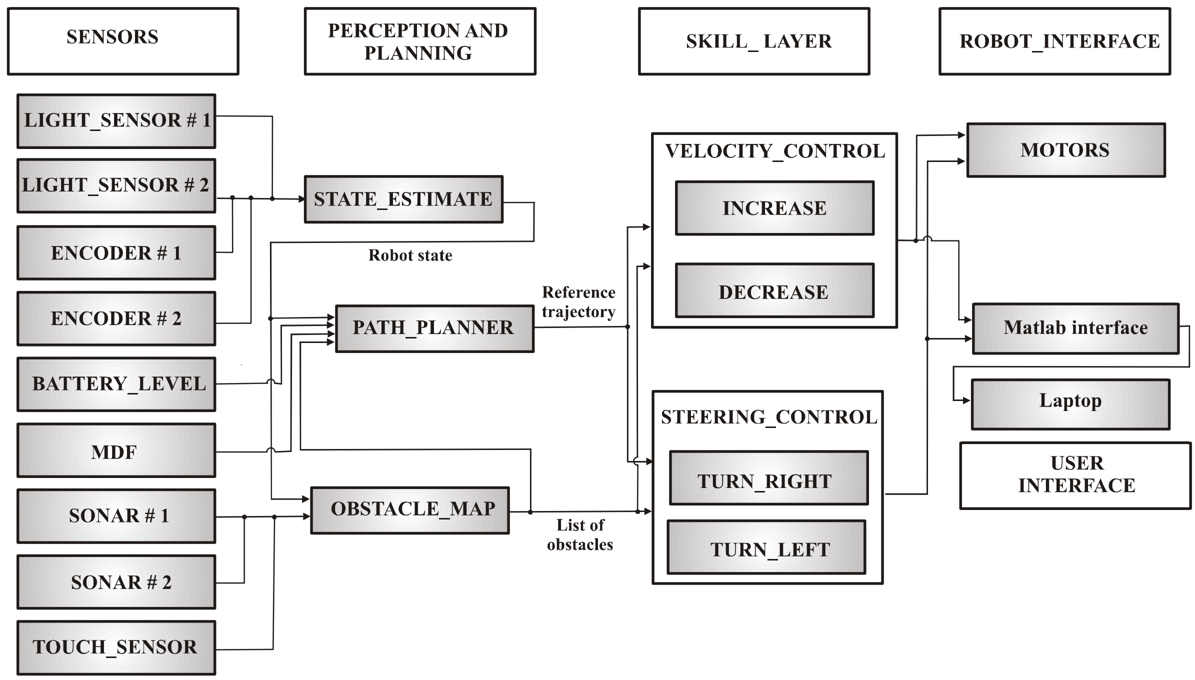

Figure 5. Proposed hybrid control architecture

The architecture consists of four distinct layers. The first is the Sensor Layer, responsible for information gathering. The second layer processes and interprets sensor information and based on that information makes decisions. As it may be seen, this layer is formed of three modules: the *State_estimate*, the *Obstacle_map* and the *Path_planner* module. The *State_estimate* module is based on Kalman Filter state estimator [17,18], providing a robot with information about its position and orientation (i.e. robot pose) in the environment. The *Obstacle_map* module detects and reports the presence of obstacles in the robot path. Finally, the *Path_planner* generates plans for online exploitation. The third layer, called Skill Layer, is responsible for fast reactions based on information provided by the Perception and Planning Layer, i.e. *Path_planner* and *Obstacle_map* modules to be more specific. The Skill Layer consists of two modules: the *Velocity_control* module generates control signals for speed control while the *Steering_control* module generates controls for change of course. Finally, the Robot Interface Layer enables control of motors. In the following part of the paper much deeper analysis of each and every module is given.

3.1 Sensor layer

For experimental setup five distinct sensors have been chosen (Fig. 6). Two ultrasonic sensors (S1 and S2), two encoders (E1 and E2), two light sensors (LS1 and LS2), one touch sensor (TS) and one battery level sensor (BL). Energy level is checked with the specific sensor that keeps track about available energy. Energy consumption is a major issue in mobile robotics and that information should be included in the decision-making process. Finally, information about transport task is given in the mission data file (MDF). These sensors are essential to intelligent behavior of the mobile robot, especially in dynamic environment like shop floor. The Sensor Layer gathers information and transfers it to the Perception and Planning Layer which is going to be explained in much more details in the subsequent part of the paper.

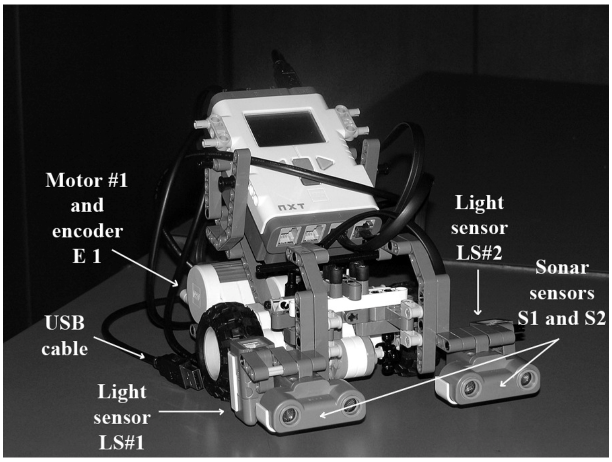

Figure 6. Mobile robot built on LEGO Mindstorms NXT Technology

3.2 Perception and planning layer

Two ultrasonic sensors (S1 and S2) should detect the presence of an obstacle in front of the robot. The *Obstacle_map* module returns answer to the following question: *Is there an obstacle in the predefined range in front of the robot?* The answer is positive if module detects the presence of obstacle and negative if there is no obstacle on the path. The obstacle detection problem could be seen as the classification problem of the machine learning [19,20]. In other words, given a data (in this particular case sonar reading) the module should provide information about the presence of obstacle.

Using the standard machine learning framework and terminology [19] we may tackle the classification problem by applying generative or discriminative learning algorithms [21]. The discriminative algorithms learn the probability $p(y|x)$ directly, where y is the label and x is the feature vector of the label, and in the classification problems these algorithms map the input vector x, where $y \in \Re^2$, into the label $y = \{0,1\}$ defined in the output space $y \in \Re^2$. For instance, if label y indicates the presence of obstacle and feature vector x indicates the sonar reading, conditional distribution $p(y|x)$ models the answer to the following question: Given a measurement x, what is the probability of obstacle presence? Therefore, for the training set S the generative algorithm should learn this mapping and for the given feature vector x return the label y.

In contrast to discriminative algorithms, the generative machine learning algorithms try to learn the inverse probability distribution, i.e. $p(x|y)$. If label y indicates the presence of obstacle in the path, then $p(x|y=0)$ models the distribution of features when there is no obstacle and $p(x|y=1)$ models the features when the obstacle is present. In addition, the class prior probability $p(y)$ has to be learned from a training set as well. Therefore, the generative learning algorithms learn how the world looks given label y (when there is the obstacle or when there is no obstacle).

Based on the previous information related to machine learning algorithms for the classification problems, the authors propose the Artificial Neural Networks (ANN) [6,22,23] for modeling this problem. As the paradigm of machine learning ANNs are able to approximate arbitrary nonlinear function [22,23] or achieve classification of the input data with predefined accuracy. For instance, the perception learning algorithm [22] achieves mapping from input vector x to discrete set $y = \{0,1\}$. In other words, the perception learning algorithm draws a line (hyperplane in a general case) between two distinct sets of data which separates these sets. Therefore, based on the input vector the perception algorithm finds the appropriate mapping.

On the other hand, the ANNs could be applied to learn distributions $p(x|y=0)$ and $p(x|y=1)$, as well as prior class distribution $p(y)$. In accordance, the ANNs could determine the value of label y based on feature vector x which in this particular problem is the vector of sonar readings.

The experimental setup based on LEGO Mindstorms NXT Technology [24] is being prepared in order to gather experimental data (sonar readings when obstacle is present and when there is no obstacle), so that valid training set could be established (Fig. 6). At this moment, feedforward neural networks [6,22,23] look suited for solving the classification problem, where the ANNs will be trained according to training set. Feedforward ANNs are known for their ability to make general conclusions based on the training set, compensate for any nonlinearities and uncertainties in the training set and to achieve reliable mapping between the input data and the desired output values. Gathering empirical data and modeling the obstacle detection problem with the ANNs should result in a fully developed module able to recognize the presence of an obstacle. In this way, the obstacle detection is being developed on the basis of empirical control strategy [25].

The third input in the *Obstacle_map* module is the output of touch sensor. Namely, if the obstacle detection algorithm fails due to imperfection of available sonars, the touch sensor is to detect and report collision with the obstacle. Finally, as one may see, the outputs of the *Obstacle_map* module are being sent to the *Path_planning* module and the Skill Layer.

Two encoders (E1 and E2) are used for purposes of odometry calculation in the *State_estimation* module. These sensors are the basis for prediction step of Kalman Filter (KF) type of the robot pose estimation [4,5,17,18,26]. Two light sensors (LS1 and LS2) are responsible for color detection of distinct features in the environment which is essential for the update step of KF and calculation of innovation vector and cross-correlation matrix between state and measurements, needed for determination of Kalman gain. At this conceptual level, the output of the *State_estimation* module is send to the *Path_planner* module.

In the proposed architecture the *Path_planner* module is being developed on the potential fields concept developed by Prof. O. Khatib from Stanford University [27]. Within this framework, the main goal the robot should reach is represented by attractive potential, while the obstacles and other not so significant objects are modelled with repulsive potential. In this way, the mobile robot is to track the generated path based on these restrictions. However, although potential field method could be applied in the real time by incrementally calculating parameters, in the proposed hybrid architecture different approach is introduced. Namely, the computational cost of online planning based on this concept is extremely high, therefore the majority of hardware capabilities will be totally concentrated towards solving this issue. Having this in mind, the mission planner (not explicitly represented in Figure 5) is to define the path the robot should take in off-line mode resulting in the mission data file – MDF. Based on the MDF the robot is to track path on-line, allowing the *Path_planner* module to modify segments of path along the way according to outputs of the *State_estimation* and the *Obstacle_map* modules. At first glance, this may be seen as an exchange of one hard problem with the other one, however, focusing on thorough off-line planning and flexible on-line tracking should enhance the overall speed of response and increase robustness of the architecture. Having this fact in mind, the mission planner is provided with transport tasks in particular the time period (MDF) and the robot is to keep up with predefined schedule of transport tasks. This fact has enormous impact since it significantly reduces the computational cost of planning. Finally, one may notice that "local" planner, i.e. the *Path_planner* module, receives information about energy consumption (provided by BL sensor) which, as a result, should have improved awareness of the IMR's performance and available energy. The outputs of the *Path_planner* module are sent directly to the Skill Layer.

3.3 Skill layer

As it is common in hybrid (deliberative/reactive) approach towards developing robotic control architectures [11] the reactive layer is called the skill layer. The Skill Layer is composed of fast reacting modules able to generate instantaneous response in real time without "thinking too much". In general, this module is responsible for obstacle avoidance, steering and velocity control or for solving any other implementation issue, where the immediate response is needed.

In the proposed hybrid architecture the Skill Layer consists of two main modules: the *Velocity_control* module and the *Steering_control* module. The *Velocity_control* module is responsible for the control of robot's speed. This module consists of two submodules, the *Increase* and the *Decrease*. According to their appellations, it is not difficult to figure out their function in the proposed architecture.

Besides the *Velocity_control* module, the robot needs the ability to steer its wheels and to change direction of travelling, so that requirements defined by the transport mission could be fulfilled. For these purposes the *Steering_control* module is proposed. As in the previous reactive module, this module is founded on two kinds of submodules: the *Turn_left* and the *Turn_right*. The function of these submodules is quite obvious.

The Skill Layer is being supplied with information generated by the Perception and Planning Layer. To be more precise, the *Path_planner* and the *Obstacle_map* modules provide the input. In this manner, the robot receives all relevant update, so that no information should be lost. The *Path_planner* module generates a set of actions that should bring the robot to the desired waypoint, i.e. the checkpoint (Fig. 7, case 1). On the other hand, the *Obstacle_map* module provides the Skill Layer with information about obstacle presence. In this manner, the Skill Layer should be able to generate a reliable response.

One may notice that unlike "standard" formulation and consequent development of behaviors in reactive paradigm [9,12], based upon individual modules, such as *move_around*, *avoid_obstacles*, *backup*, *approach_goal*, etc. in the proposed architecture the Skill Layer is founded on the basis of their mixture. In other words, there is no explicit module called *avoid_obstacles*, *approach_goal* or *follow_the_path*, instead the individual submodules in the Skill Layer should generate response according to stimuli. Therefore, whether the robot should perform the obstacle avoidance maneuver or approach goal point, the Perception and Planning Layer is to recognize the action and send it to the Skill Layer.

At this point of the research the Skill Layer is responsible for the path following and obstacle avoidance. However, unlike "conventional" software solutions based on *if-then* rules (as in subsumption architecture) or vector calculation (as in motor schemas) authors propose the use of the Artificial Neural Networks for their assemblage. Namely, modeling this problem with the ANNs by gathering empirical data, based on experimental setup, should result in improved responses of modules and control of the robot. In the following part of the paper the proposed methodology will be provided.

As it may be seen in Figure 5 the *Path_planner* module provides the Skill Layer with information of reference trajectory the mobile robot should follow. For instance, that information could be defined with vector in the *distance to waypoint-course to waypoint* space. Therefore, at this point, the *Path_planner* module outputs the following vector $p = \{\rho\ \varphi\}$, where ρ is the distance to waypoint and φ is needed change of the course. On the other hand, the *Obstacle_map* module outputs the list of obstacles and the distance between the robot and obstacle(s). Put it differently, this particular module outputs the following vector $o = \{d\ l\ r\}$, where d is the distance, while components l and r define the obstacle presence on the left and on the right side of perceptual field respectively (Fig. 7, case 2, a) and b)).

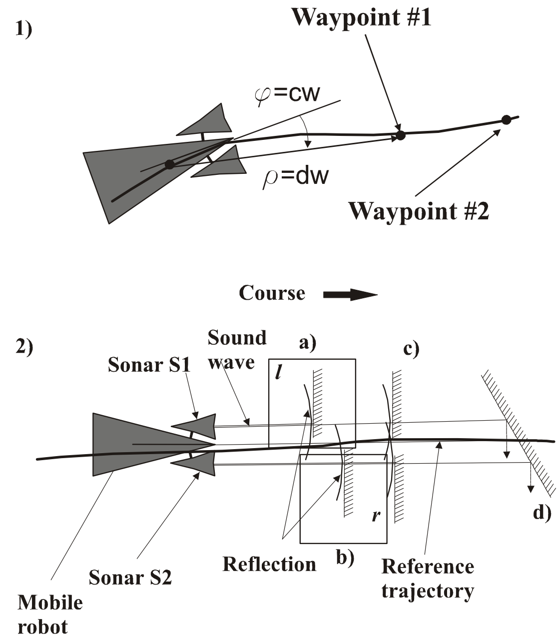

Figure 7. Sonar measurements

One may notice that sonar sensors S1 and S2 search for obstacles right in front of the robot, neglecting the possible presence of obstacles perpendicular to the reference trajectory. In this way, the information flow would be enhanced since the presence of obstacles outside the perceptual field has a minor influence (or no influence at all) on the planned path. If these obstacles were static, their influence has been analyzed and taken into account during the off-line path planning. On the other hand, if these obstacles were dynamic ones, in the manufacturing environment these obstacles could be human operators or other mobile robots. Either way, workers would avoid collisions with the IMRs, while the IMRs would be programmed and learned not to collide with the other IMRs.

Based on previous analysis, one may conclude that input vector i in the Skill Layer, defined in $\mathbf{R}^5$ space, is composed of the following components: $i = \{\rho\ \varphi\ d\ l\ r\}$. This vector is sent to both modules of the Skill Layer: the *Velocity_control* module and the *Steering _control* module. Two distinct feedforward neural networks will be used for information processing and generation of the output. The first one is the Velocity Control Neural Network (VCNN), responsible for generation of speed change, while the other one is the Steering Control Neural Network (SCNN), responsible for change of the robot's course. The output of the VCNN network is scalar value Δv that defines a new value of translational velocity. Similarly, the output of the SCNN is defined with vector $s = \{\Delta l\ \Delta r\}$, where Δl represents control input to the left motor and Δr is control input to the right motor, so that change of the course could be performed. In this manner, based on the information provided by the Perception and Planning Layer, the future actions of the robot should be defined. However, synchronization of these two neural networks is a major issue. Namely, simultaneous processing of inputs, with both neural networks working on-line, is not possible, merely because of operating system's inability to perform multi-tasking or multi-threading. If this were possible, it would be quite easy to perform computation of the outputs. This software issue could be solved by defining the importance of motor control. For instance, the output of the VCNN could be declared as more important information than the output of the SCNN. In this manner, the VCNN output will be the first one sent to motors. Having sent these outputs, the SCCNN will send the rate of steering change. Justification for introduction of action importance is empirical one: when obstacle is detected the first thing robot should do is to decrease the velocity and then to change its course. Nevertheless, one may easily notice that vice versa approach (the SCNN output then the VCNN output) may hold as well.

Yet, another way of tackling synchronization problem is to implement additional neural network. Namely, as in the previous case, first the VCNN's output will be calculated, and after that the SCNN's output. Finally, the new neural network will receive the input vector, and generate motor controls. This solution of the synchronization issue resembles the Modular Artificial Neural Networks (MANN) approach, where output of n neural networks (one may say expert neural networks as well) is being sent to a new network for improved decision making. Either way, the decision whether to choose the first or the second approach for synchronization will be made during experimental process.

4. DISCUSSION AND EXPERIMENTAL SETUP

In the proposed architecture there is no explicit modeling of robot behaviors. Namely, by modeling more general phenomena specific behaviors like *avoid_obstacles* should emerge, ensuring existence of the emergent behaviors [16]. In this way, the ANNs should capture specifics of each behavior and result in much improved architectural output. Therefore, by applying machine learning and empirical control, we let the mobile robot learn non-modelled behaviors needed for exploitation on daily level.

An interesting problem may arise when *Obstacle_map* module, based on the interpretation of sonar readings, detects the presence of obstacles on both sides of sonar perceptual field (Fig. 7, case 2. c)). Namely, in this particular situation the initial architectural setup is not able to solve this problem. Therefore, in order to tackle this issue, additional module, or to be more precise, submodule of *Path_planner* could be added. This module is to be developed on the basis of the action selection algorithm presented in Figure 8.

The new action selection algorithm starts by checking whether the obstacle is present in both sides of perceptual field. If this is the case, the robot will collect data, classify the environment and based on that information generate a brand new plan that will enable robot to go around the obstacle.

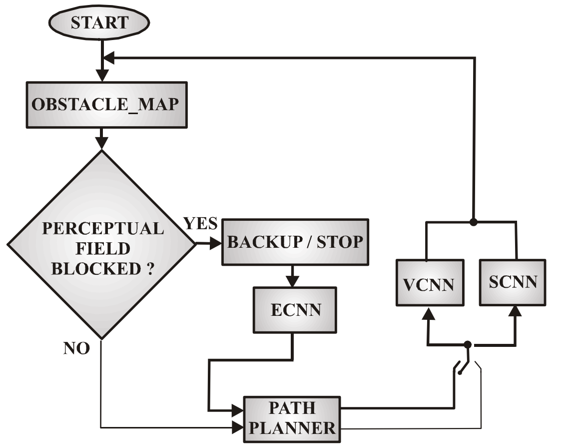

Figure 8. Action Selection Algorithm

For these purposes, the environment classification has to be enabled. This situation could be solved by introducing Environment Classification Neural Network (ECNN) that should take sonar readings and achieve classification, so that the robot would know whether it stands next to left wall, right wall, hallway, etc. (Fig. 9).

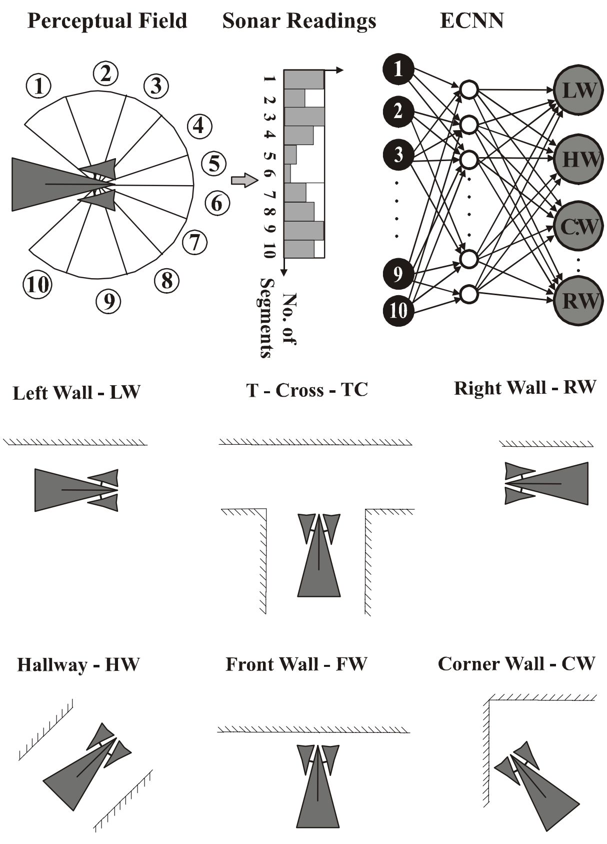

Figure 9. Environment Classification Neural Network

A similar idea of the environment classification based on multilayer perception could be seen in [28] but with two major distinctions. Firstly, in [28] the environment classification is performed in real time, unlike this particular implementation of the ECNN which assumes that robot has stopped moving, all due to available resources for experimental setup. The second distinction is related to training set necessary for adjustments of weights between neurons. Namely, in [28] the ECNN was trained with simulated sonar readings. The readings had been gathered and afterwards the ECNN was trained with added noise. Unlike this approach, the authors propose the empirical approach towards gathering sonar data that would enable more reliable and more precise training set necessary for adjusting of the weights between neurons.

Figure 7 (case 2, d)) shows the unreliableness of sonars and dissipation of sound waves. If the surface is not perpendicular to the direction of wave's travelling, the sonar wave will be dissipated. Therefore, this particular situation results in the reflection of sound waves proportional to the incoming angle, resulting in sonar misreading. This situation, so common in "the real world", depicts why empirical data has to be gathered and why the concept of machine learning should be implemented. Applying learning algorithms and the ANNs should enable robot to recognize the presence of obstacle although at first glance the presence of obstacle was disregarded.

The proposed architecture could be extended in several ways. The first one is related to the perception ability. For instance, information about the collision with objects (provided by touch sensor) could be sent to the Skill Layer directly as well, providing the VCNN and the SCNN modules with additional parameter needed if collision with the obstacle happens. Touch sensor is to send the information to the Skill Layer significantly improving the overall response. The second extension is related to map making ability [26]. When the robot is set to run for the first time, the human operator should control robot and "show" the specifics of the environment. On the first run, this module would generate a map and use this map for tasks in the near future. At the end of the first run, the map will be stored and used for navigation purposes in the future, while the map building module will "become" state estimation module. Needless to say that during exploitation this module could be started at any time and set to work mode if unseen feature of the environment shows up. At this stage of architectural development the *State_estimate* module was developed exclusively for localization purposes. However, the module itself could be extended, so that map building could be achieved.

For the time being, a new hybrid architecture will be tested on mobile robot built on Lego Mindstorms NXT technology [24] operating in the experimental working environment (Fig. 6). Due to limited resources of Lego's hardware and software, the laptop will serve as the main computation unit, while communication between the robot and the laptop will be achieved with Matlab [29] via USB or Wireless protocol. In this way, the laptop will perform all necessary computation on-line and send commands to the robot knowing that embedded computation is not achievable at this point of research. On the other hand, although computational power will be improved significantly, all due to Matlab and laptop capabilities, the authors are aware of possible delays in communication and influence on experimental process. However, if architecture performs well on this (admittedly) limited experimental platform, then we are one step closer to embedded computation.

5. CONCLUSION

In this paper, the new hybrid architecture for mobile robot navigation and exploitation in a manufacturing environment was presented. The architecture is based on hybrid approach towards developing intelligent robotic systems, so that fruits of both Hierarchical and Behavior-based styles should be captured. The initial architectural setup is founded on gathering of empirical data and implementation of machine learning through development of reliable models of artificial neural networks for obstacle detection, behavior generation and environment classification.

The use of ANN paradigm for behavior modelling should result in improved capabilities of a mobile robot and provide much needed robustness to sensor failures. Neural networks will be trained by empirical data gathered in the experimental process. Therefore, the output should capture the influence of significant parameters and achieve generalization. The experimental setup, which is being prepared at the moment, should point out the advantages and disadvantages and ultimately verify the usability of architecture and proposed approach towards development of the IMR's as intelligent agents.

Finally, these issues have been analyzed from the perspective of mobile robot implementation in a manufacturing environment as an integral element of transport system, which in turn should result in improved performance of manufacturing systems and processes.

ACKNOWLEDGMENT

This paper is a part of the research called: *Flexible Automation and Implementation of Intelligent Manufacturing Systems in Sheet Metal Production*, being financed by The Serbian Government, The Ministry of Science and Technological Development under the grant TR-14031.

The authors would like to thank their partners in SAGA Company – Belgrade (www.saga.co.yu) as well for their help and much needed support achieved by donating Hewlett-Packard notebook PC HP Compaq 6820s. Their help is gratefully acknowledged.

REFERENCES

[1] Wright, P.K.: *21st Century Manufacturing*, Prentice Hall, Upper Saddle River, 2000.

[2] Groover, M.P.: *Automation, Production Systems, and Computer-Integrated Manufacturing*, Prentice Hall, Upper Saddle River, 2001.

[3] Kopacek, P.: Intelligent manufacturing: Present state and future trends, Journal of Intelligent and Robotic Systems, Vol. 26, No. 3-4, pp. 217-229, 1999.

[4] Miljković, Z., Vuković, N. and Babić, B.: Mobile robot localization in a manufacturing environment, in: *Proceedings of the 3rd International Conference on Manufacturing Engineering (ICMEN 2008)*, 01-03.10.2008, Kallithea of Chalkidiki, Greece, pp. 485-494.

[5] Vuković, N., Miljković, Z., Babić, B. and Čović, N.: Industrial mobile robots in intelligent manufacturing systems, Tehnika – Mašinstvo, Vol. 57, No. 4, pp. 11-18, 2008, (in Serbian).

[6] Miljković, Z.: *Systems of Artificial Neural Networks in Production Technologies*, Faculty of Mechanical Engineering, Belgrade, 2003, (in Serbian).

[7] Murphy, R.R.: *Introduction to AI Robotics*, MIT Press, Cambridge, Massachusetts, 2000.

[8] Fukuda, T. and Arakawa, T.: Intelligent systems: Robotics versus mechatronics, Annual Reviews in Control, Vol. 22, pp. 13-22, 1998.

[9] Arkin, R.: *Behavior-Based Robotics*, MIT Press, Cambridge, Massachusetts, 1998.

[10] Russell, S.J. and Norvig, P.: *Artificial Intelligence: A Modern Approach*, Prentice Hall, Upper Saddle River, 2002.

[11] Gat, E: Three-layered architectures, in: Kortenkamp, D., Bonasso, R.P. and Murphy, R. (Eds.): *Artificial Intelligence and Mobile Robots: Case Studies of Successful Robot Systems*, AAAI Press, Menlo Park, pp. 195-210, 1998.

[12] Brooks, R.A.: A robust layered control system for a mobile robot, IEEE Journal of Robotics and Automation, Vol. 2, No. 1, pp. 14–23, 1986.

[13] Irobot Corporation, http://www.irobot.com.

[14] Thrun, S. et al.: Stanley: The robot that won the DARPA grand challenge, Journal of Field Robotics, Vol. 23, No. 9, pp. 661-692, 2006.

[15] Urmson, C. et al.: Autonomous driving in urban environments: Boss and the urban challenge, Journal of Field Robotics, Vol. 25, No. 8, pp. 425-466, 2008.

[16] Pfeifer, R. and Scheirer, C.: *Understanding Intelligence*, MIT Press, Cambridge, Massachusetts, 2001.

[17] Kalman, R.E.: A new approach to linear filtering and prediction problems, Transactions of the ASME – Journal of Basic Engineering (Series D), No. 82, pp. 35-45, 1960.

[18] Bar-Shalom, Y., Rong Li, X. and Kirubarajan, T.: *Estimation with Applications to Tracking and Navigation – Theory, Algorithms and Software*, John Wiley & Sons, New York, 2001.

[19] Mitchell, T.M.: *Machine Learning*, McGraw Hill, New York, 1997.

[20] Alpaydin, E.: *Introduction to Machine Learning*, MIT Press, Cambridge, Massachusetts, 2004.

[21] Ng, A.Y. and Jordan, M.: On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes, in: Dietterich, T.G., Becker, S. and Ghahamani, Z. (Eds.): *Advances in Neural Information Processing Systems 14, Vol. 1*, MIT Press, Cambridge, Massachusetts, pp. 841-849, 2002.

[22] Haykin, S.: *Neural Networks: A Comprehensive Foundation*, Prentice Hall, Upper Saddle River, 1998.

[23] Dreyfus, G.: *Neural Networks – Methodology and Applications*, Springer-Verlag, Berlin, 2005.

[24] http://mindstorms.lego.com.

[25] Miljković, Z. and Babić, B.: Empirical control strategy for learning industrial robot, FME Transactions, Vol. 35, No. 1, pp. 1-8, 2007.

[26] Vuković, N. and Miljković, Z.: Extended Kalman filter in autonomous mobile robot localization and mapping, Bulletin of the Transilvania University of Brasov – Series A, Vol. 15, No. 50, pp. 435-444, 2008.

[27] Khatib, O.: Real-time obstacle avoidance for manipulators and mobile robots, in: *Proceedings of the 1985 IEEE International Conference on Robotics and Automation*, 1985, St. Louis, pp. 500-505.

[28] Na, Y.-K. and Oh S.-Y.: Hybrid control for autonomous mobile robot navigation using neural network based behavior modules and environment classification, Autonomous Robots, Vol. 15, No. 2, pp. 193-206, 2003.

[29] http://www.mathworks.com.

НОВА ХИБРИДНА УПРАВЉАЧКА АРХИТЕКТУРА НАМЕЊЕНА ЗА ЕКСПЛОАТАЦИЈУ ИНТЕЛИГЕНТНИХ МОБИЛНИХ РОБОТА У ПРОИЗВОДНОМ ОКРУЖЕЊУ

Најдан Вуковић, Зоран Миљковић

У раду је приказана нова хибридна управљачка архитектура намењена за експлоатацију и навигацију интелигентних мобилних робота у технолошком окружењу. Архитектура је базирана на емпиријском управљању и имплементацији концепта машинског учења у виду развоја система вештачких неуронских мрежа за потребе генерисања интелигентног понашања мобилног робота. За разлику од конвенционалне методологије развоја интелигентних мобилних робота, предложена архитектура је развијена на темељима експерименталног процеса и имплементације система вештачких неуронских мрежа за потребе генерисања интелигентног понашања. Предложена методологија развоја и имплементације интелигентних мобилних робота треба да омогући несметану и поуздану експлоатацију али и робустност у погледу генерисане управљачке команде, као одговора робота на тренутно стање технолошког окружења.