IJST, Transactions of Mechanical Engineering, Vol. 36, No. M1, pp 25-40
Printed in The Islamic Republic of Iran, 2012

TOWARDS IMPLEMENTATION AND AUTONOMOUS NAVIGATION OF AN INTELLIGENT AUTOMATED GUIDED VEHICLE IN MATERIAL HANDLING SYSTEMS*

B. BABIĆ[1],** Z. MILJKOVIĆ[1], N. VUKOVIĆ[2] AND V. ANTIĆ[3]

[1]University of Belgrade, Faculty of Mechanical Engineering, Kraljice Marije 16, 11120 Belgrade, Serbia
Email:bbabic@mas.bg.ac.rs
[2]Innovation Center of Faculty of Mechanical Engineering, Belgrade, Serbia
[3]Center of Applied Mathematics and Electronics, Belgrade, Serbia

Abstract– Automated Guided Vehicles (AGV) have been a conventional solution and choice made by many manufacturing enterprises as means for Flexible Material Handling Systems (FMHS). In recent years, a considerable number of these vehicles have been installed on shop floors worldwide, effectively proving the usefulness of material handling systems. However, the increasing complexity of demand as well as a need for "make to order" rather than "make to stock" policy implies usage of more intelligent material handling solutions. This paper discusses the usage of an intelligent AGV as means for FMHS which should have Intelligent Material Handling System (IMHS) as the final outcome. This paper presents the experimental results of hybrid robotic control architecture. To evaluate the performance of the architecture, a mobile robot built on LEGO® Mindstorms NXT technology was used. Some of the architectural modules are based on the implementation of Artificial Neural Networks in order to achieve the needed robustness in exploitation. Then, through simulation using AnyLogic® 6 software, the performance in terms of manufacturing system effectiveness and workstation utilization is analyzed. Based on the experimental results, the proposed IMHS can have significant advantages over conventional material handling systems.

Keywords– Flexible material handling system, intelligent automated guided vehicle, artificial neural networks, discrete event simulation

1. INTRODUCTION

A flexible manufacturing system (FMS) is a fully integrated manufacturing system consisting of computer numerically controlled (CNC) machines, connected by an automated material handling system, all under the control of a central computer [1]. Flexible manufacturing systems are crucial for modern manufacturing to enhance productivity involved with product proliferation. As one of the critical components of the FMS, the flexible material handling system (FMHS) plays a strategic role in the implementation of the FMS [2, 3].

Material Handling Systems (MHS) represent automated transport of raw materials, partially manufactured products and goods between different locations of manufacturing systems [4]. Conventional transport solutions are based on powered and non-powered industrial trucks, conveyer belts, vertical conveyors, material handling robots and Automated Guided Vehicles (AGVs) [4, 5]. Although AGV technology has been transformed from predetermined guide paths [6-14] to recent advances [4, 15, 16],

*Received by the editors July 29, 2010; Accepted September 29, 2011.
**Corresponding author

there are still open fields for the research. If one considers the technological development that the field of mobile robotics has achieved, it is easy to see where some aspects of AGV research should be directed. By implementing these results [17-19] the AGV based transport could become faster, efficient and more reliable. The first effort in this direction is given in [20].

In this paper we present the experimental results of hybrid (robotic) control architecture for autonomous navigation of automated guided vehicles in a manufacturing environment. To evaluate the performance of architecture, a mobile robot built on LEGO® Mindstorms NXT technology was used. An experimental model of the existing manufacturing environment was developed and the results of the AGV implementation based on the proposed architecture were presented. Furthermore, in AnyLogic software we developed the model of the manufacturing environment and analyzed how implementation of advanced AGV based transport can improve its efficiency. It is important to stress that the empirical data is used for the simulation study. Data was recorded while analyzing and measuring the productivity of each machine tool. Simulation studies are useful for analysis of various processes of manufacturing enterprises [21].

Studies of AGV design and control [22, 23] point out that a new generation of AGV will be necessary to handle tasks in various industrial applications in years to come. Both studies emphasize the necessity for more intelligent vehicles able to tackle problems being imposed with demands for more effective, reliable, faster, and efficient transportation systems based on this technology. It is evident that these requirements imply usage of artificial intelligence, as well as methods and algorithms developed by a research community in the field of mobile robotics. This should result in more intelligent vehicles able to flexibly adapt to changes in a working environment while performing the transportation task.

This paper is structured as follows. After discussing related work in the following section, in the third section we will present the layout of the manufacturing environment used in this research. Our approach towards developing robotic control architecture is given in the fourth section. Finally, the fifth part provides experimental and simulation results demonstrating the advantages of our approach.

2. RELATED WORK

AGV based transport is a well-studied field. There are a number of sources reporting successful implementation of AGVs for ship container transport in harbours [24, 25], industrial transport [4, 16] and even hospital environment [26]. Kelly et al. [15] developed an AGV transport system with camera able to operate in an environment without a supporting infrastructure. They noticed that retrofitting (existing) industrial trucks by computer vision and laser detection and ranging can reduce dependence on guidance structure. In their research [27, 28] they focused on various aspects of AGV design, but no reference is made towards development of robotic control architecture.

In the study of AGV much research effort is devoted to the problem of routing, which represents a vehicle's ability to make decisions along the guidance path in order to select optimum route to destination [4]. Scheduling and routing seemed trivial, but as the number of installed AGVs became larger this issue gained in importance [29]. Recent advances in this domain have proposed disjunctive graph and memetic algorithm for simultaneous scheduling of machines and automated guided vehicles [30]. Nishi et al. [31] provided a bilevel decomposition algorithm for simultaneous production scheduling and conflict-free routing for AGVs by decomposing original problem into upper and low-level subproblems. A similar idea for decomposition of original problem is found in [32]. On the other hand, Angeloudis *et al.* [33] developed a method for uncertainty penalization and report that the proposed method outperforms known approaches and heuristics. Berman et al. [34] developed methodology for evaluation of AGVs control based on their long-term research [35, 36]. Multi-vehicles approach to industrial transport problem requires analysis of other issues as well [36, 37].

This paper differs from the aforementioned contributions. Our ongoing research focuses on implementation of advanced methods and algorithms developed by the community in the field of mobile robotics for transport purposes within the manufacturing environment. One recent published result [4] is similar to our approach. They considered the implementation of flexible AGV based on industrial forklift truck. To build the flexible AGV the authors reported that their system is easily configured, easily commanded, able to navigate autonomously and able to operate in partially structured environments [4]. In a way, the aforementioned abilities can easily become standard conditions that the flexible AGV system needs to fulfill. Our work differs from their contribution in several ways. First of all, in this research we focused on the building and testing of the proposed navigation architecture in an experimental environment. We developed hybrid robotic control architecture for navigation in a manufacturing environment and present the first experimental results. The hybrid nature of the architecture allows both deliberative processes for planning purposes and reactive behaviours for motor control. Synergy of reactive and deliberative approaches has been recognized as a better solution [38-40]. Furthermore, by using feedforward artificial neural networks for some of the individual architectural modules the robustness of the architecture is improved. Artificial neural networks are able to achieve complex nonlinear mappings and operate in real time with a high degree of accuracy. The second distinction relates to building a simulation model of the analyzed manufacturing enterprise. Simulation study shows whether the effectiveness of the manufacturing system and work station utilization can be improved.

3. MANUFACTURING ENTERPRISE

This part of the paper briefly describes the layout used in the simulation and experimental process. This particular manufacturing enterprise is in the business of manufacturing sheet metal and their assemblage in products. The layout may be seen in Fig. 1.

The authors designed the layout two years ago. The layout is designed in a circular shape to enhance the flow of material and to increase productivity. The existing environment has nine machines (Table 1). The warehouses are marked as W1, W2, W3, W4 and W5. From the standpoint of material transport, the important warehouses are W1 (assembling and packaging) and W2 (final products). The local warehouses (W4 and W5) are located in the vicinity of machines (Fig. 1). Powered and non-powered industrial trucks perform the material transport tasks. The main research goal at this stage is to see whether it is possible to improve the flow of material and consequently increase efficiency by implementing intelligent material handling system (IMHS). Manufacturing process starts with the machine M1, where panels of the sheet metal are being sheared into the smaller panels. Depending on the product being produced, the smaller panels are sent to be punched or blanked. Finally, the material is sent to W1 where assembling and packaging are performed. The final products are transported to the warehouse W2. As expected, transportation paths are not the same for all products. The paths could be diverse depending on the parts being manufactured and materials needed.

4. CONTROL ARCHITECTURE

To implement mobile robots as a means of transport the particular control architecture has been proposed [41]. The architecture consists of four distinct layers (Fig. 2.). The first one is the Sensor Layer responsible for information gathering. The second layer processes and interprets the sensor information and makes decisions. The third layer generates fast reactions according to the information provided by the second layer. Finally, the Robot Interface Layer enables the control of motors. The architecture is designed regarding functions for the intelligent agent [42].

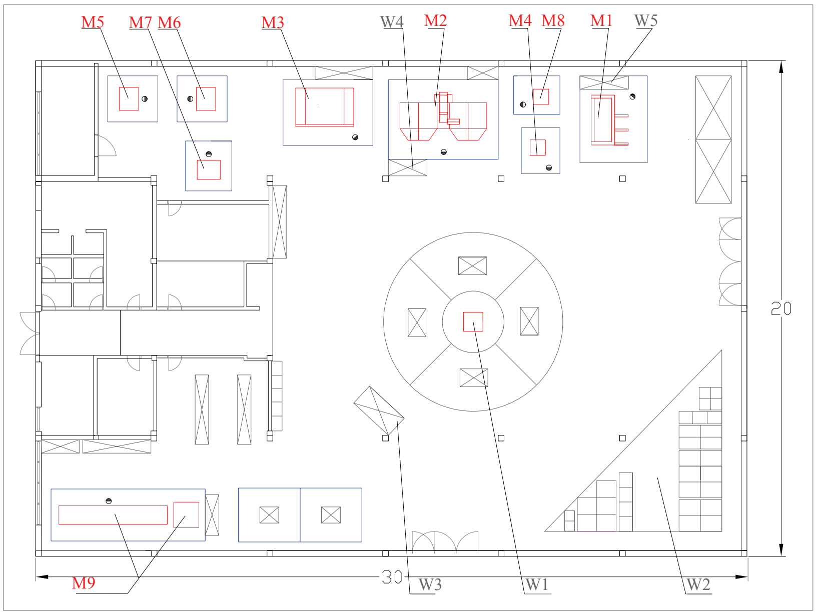


Fig. 1. Layout of the existing manufacturing environment

Table 1. List of machines in the manufacturing enterprise

Machine symbol	Description
M1	Shearing machine
M2	CNC punch press for punching and blanking
M3	Hydraulic punch press
M4	Punch press for punching and blanking
M5 and M6	Pillar drill (bench drill)
M7	Circular saw
M8	Whetting machine
M9	Line for machining parts made of cooper

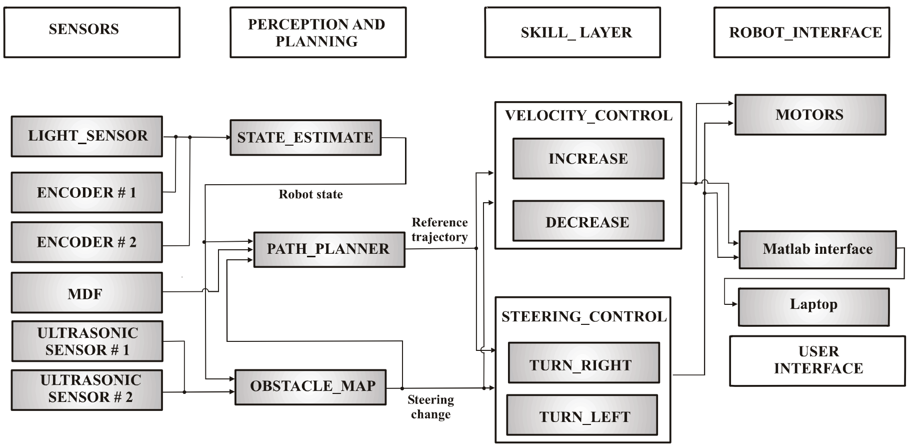


Fig. 2. Overview of the architecture

The second layer consists of three modules: *State_estimate*, *Obstacle_map* and *Path_planner* module. *State_estimate* module provides a robot with information about its pose. *Obstacle_map* module detects and reports the presence of obstacles in the path, while *Path_planner* generates plans for online exploitation. The information about a transport task is given in the mission data file (MDF) that takes the production plan on a daily level and sends them to *Path_planner*. The output of the *State_estimation* module is sent to the *Path_planner* module. The Skill Layer consists of one module: the *Steering_control* module generates controls for the change of the robot's course. The Skill Layer is supplied with information generated by *Path_planner* and *Obstacle_map* modules. This module will be in our research focus in the near future. The following part of the paper introduces individual modules and provides main ideas implemented during the experimental process.

a) State estimation

The mobile robot localization is the problem of determining the robot pose relative to the given map of the environment [43]. The problem itself assumes that the mobile robot was given the map and, relative to that particular map, the robot is supposed to estimate its position and orientation. There are two main approaches towards solving localization problem. The first one is global localization problem, the problem where the robot knows the map of the environment but its initial position and orientation are not known. The second one is the local localization problem (initial position and orientation are known), often referred in the research community as position tracking. Our implementation is based on the second approach.

In the rich field of literature related to mobile robot localization, a number of methods have been proposed as a solution [44-53]. In this paper the state estimation module is based on the Extended Kalman Filter (EKF) [54-56]. In Kalman Filter (KF) setup the state transition model is given as:

$$x_t = F_t x_{t-1} + B_t u_t + w_t \tag{1}$$

where x_t and x_{t-1} is the state vector at time instants t and $t-1$ respectively, while u_t is the control vector at time t. F_t is a square matrix of size $n \times n$ (n is the dimension of the state vector x_t) and B_t is of size $m \times n$ (m is the dimension of the control vector u_t). Finally, w_t is the zero mean uncorrelated motion noise with the Gaussian distribution and covariance matrix Q_t, i.e. $w_t \sim N(w_t;0,Q_t)$. The measurement model is defined as follows:

$$z_t = H_t x_t + v_t \tag{2}$$

Here, H_t is a matrix of size $k \times n$, where k is the dimension of the measurement vector z_t, and v_t is the measurement noise with the Gaussian distribution with zero mean uncorrelated noise and known covariance matrix R_t, i.e. $v_t \sim N(v_t;0,R_t)$. Furthermore: $E[w_k w_j^T] = Q_k\,\delta$; $E[v_k v_j^T] = R_k\,\delta$; $E[v_k w_j^T] = 0$, where δ is the Kronecker delta function ($\delta = 1$ if $k = j$ and $\delta = 0$ if $k \neq j$). KF is implemented in two steps, the prediction step and the update step. The prediction step predicts *a priori* the state of the robot represented with expected valued μ_t and covariance matrix $\sum_t$ of assumed Gaussian distribution according to the state transition model. In the update step, the information gathered from measurements is incorporated by calculating Kalman gain and innovation vector. Although KF is one of the most widely used probabilistic techniques, it is not capable of estimating robot pose with predefined accuracy due to linearity assumption. Therefore, Extended Kalman Filter (EKF) was introduced [54-56]. EKF extends nonlinear models of state transition model and measurement model via Taylor series expansion. Let x_t be the state vector, and Σ_t covariance matrix at time t, and let $g(x_{t-1}, u_t)$ and $h(x_t)$ be the nonlinear functions that define the state transition model and the measurement model respectively. Then the EKF equations for mobile robot localization are as follows:

$$G_t = \frac{\partial g(x_{t-1}, u_t)}{\partial x_{t-1}} \quad ; \quad V_t = \frac{\partial g(x_{t-1}, u_t)}{\partial u_t} \tag{3}$$

$$\bar{\mu}_{t-1} = \mu_{t-1} + G_t(x_t - \mu_{t-1}) \quad ; \quad \bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + V_t M_t V_t^T \tag{4}$$

$$S_t^i = H_t^i \bar{\Sigma}_t [H_t^i]^T + R_t \quad ; \quad K_t^i = \bar{\Sigma}_t [H_t^i]^T [S_t^i]^{-1} \tag{5}$$

$$\bar{\mu}_t = \mu_t + K_t^i (z_t^i - \hat{z}_t^i) \quad ; \quad \bar{\Sigma}_t = (I - K_t^i H_t^i)\bar{\Sigma}_t \tag{6}$$

G_t and V_t are Jacobians of motion model with respect to the state vector x_{t-1} and control vector u_t respectively. H_t is measurement Jacobian with respect to state vector x_{t-1} ($H_t = \partial h(x_t)/\partial x_t$). In the second line the prediction step is taken. In the third line of the algorithm, first the uncertainty of measurement is determined and after that the Kalman gain. Finally, the update step is taken in the fourth line of the algorithm. The presented algorithm recursively calculates the robot pose and the uncertainty.

The sensor model of a mobile robot perception uses feature based maps. The feature vector is defined as: $f(z_t) = [x_t^n \ y_t^n \ s_t^n]^T$, where x_t^n and y_t^n stand for the x and y coordinates of feature n defined in global frame, and s_t^n represents its signature. In this experimental setup, the signature of the feature has been defined as its colour. The light sensor is used as a sensor for detection of light change.

b) Path planning

Planning the robot's future actions is done exclusively for path finding according to the plan of transport, which provides the information regarding materials needed to be transported as well as transportation start points and goal points (machine tools and warehouses). MDF module in the sensor layer takes that information and generates the optimal path the robot should take in accordance with the requirements imposed by the daily production schedule. At this stage of the research, *Path_planner* module and MDF module are the same modules. For purposes of path planning the A* search algorithm is implemented [57, 58]. A* algorithm searches for the shortest path by finding the minimum of the following cost function:

$$f(n) = g(n) + h(n) \tag{7}$$

Here *g(n)* is path-cost function, which represents the cost of going from node *i* to node *j*, while *h*(*n*) is heuristic, which determines the distance from node *j* to the goal. A* search is guaranteed to find the optimal path if heuristic *h*(*n*) is optimal. The experimental model of manufacturing environment has been discretized into a *100 x 150* grid of pixels (*1x1* [cm] is dimension of pixel) and eight-point connectivity has been adopted.

c) Steering control neural network

Neural networks have received considerable attention recently in the various fields of engineering research, all due to their ability to approximate a large class of functions [59,60]. Neural network provides a designer the opportunity to model complex engineering problems where "conventional" solutions have limited capabilities.

In order to achieve motor control one module has been developed for control of steering the robot's wheels. In the experimental setup, Steering Control Neural Network (SCNN) has been developed on the basis of Generalized Regression Neural Network (GRNN) [59]. This particular neural network is formed of two layers. The first one, called the hidden layer, consists of Radial Basis Functions (RBF) performing nonlinear transformation of the input. The second layer performs weighted summation of the first layer's outputs. This neural network is suitable for function approximation, drawing the function estimate directly from the training data. In the following part of the paper the basics of this neural based regression will be introduced. Nonlinear regression function is given as follows:

$$y_i = f(x_i) + \varepsilon_i , \qquad i = 1,2,...,N \tag{8}$$

Here, x_i is the input vector ($x \in \Re^m$), y_i is the observable ($y \in \Re$) and $f(x_i)$ is unknown regression function. ε_i represents white noise with zero mean and variance σ^2. Moreover:

$$E[\varepsilon_i] = 0 \qquad ; \qquad E[\varepsilon_i \varepsilon_j] = \begin{cases} \sigma^2 & i = j \\ 0 & otherwise \end{cases} \tag{9}$$

The estimate of regression function $f(x_i)$ is based on expected values of model output y near a point x. By applying this concept the regression of y on x is given as:

$$f(x) = E[y \mid x] = \frac{\int_{-\infty}^{\infty} y\, f_{x,y}(x,y)dy}{\int_{-\infty}^{\infty} f(x,y)dy} \tag{10}$$

where density $f(x,y)$ is generally not known and must be estimated from a sample of observations of y and x:

$$\hat{f}(x,y) = \frac{1}{N\, h^{m+1}} \sum_{i=1}^{N} K(\frac{x - x_i}{h}) K(\frac{y - y_i}{h}) , \qquad x \in \Re^m,\ y \in \Re \tag{11}$$

where h is the parameter controlling the size of the RBF units $K(\cdot)$, also called kernels. The adjustable parameters of this network are the location of basis functions, the width of the kernel (the spread, i.e. parameter h), the shape of the receptive field and the linear output weights. GRNN is a normalized RBF network with a centred hidden unit at each training example.

Having introduced the basic concept of generalized regression neural network, the following part of the paper presents the experimental process and the results of training. The experimental process was performed in the following manner. The desired change of the steering angle was declared as the input while the motor commands were regarded as the output. The robot was given the command to change its orientation for a particular angle relative to its initial orientation. The angles used in the experimental process are: ±15°; ±30°; ±45°; ±60°; ±90°; ±120°; ±150°; ±180°. Then, the motor command that generates the required steering for the issued change of the orientation was recorded. Fig. 3 shows the training set (a) and the training results (b). Dots represent the training set while the lines are results of the learning process. Each line depicts the learned hypothesis for different choice of parameter h which controls the spread of the kernel function $K(\cdot)$. For larger values of this parameter the estimated function becomes smoother. The arrow in Fig. 3a shows functions with larger numerical values of spread parameter. The best results were obtained for the $h=1.2$. Fig. 3b presents results of the training process for various values of parameter h (1…5 with increment of 0.2). Values larger than five have been disregarded in the training process since these values tend to generate the network's response, which is not the desired one. The network's performance was tested at the following points t = (-170, -130, -80, -25, 25, 80, 130, 170) in [°].

It is important to stress that the generalized regression neural network is able to approximate the unknown function with the zero value of the mean squared error, which is the case in this example. Based on the previous analysis it is obvious that GRNN is able to perform desired mapping between the input space and the output space.

d) Colour recognition

To implement EKF algorithm for the robot's localization it was essential to enable the robot with the ability to distinguish colour of basement and colour of landmarks. As it may be seen, the experimental manufacturing environment is painted in white, while the landmarks are painted in black colour – Fig. 4a (the black rectangles are landmarks). For these purposes the following experimental procedure has been applied so that training set could be established. The mobile robot was placed in the environment and given the command to move. While the robot was moving the light sensor gathered readings. Having moved for some time, the operators declared what the robot had been driven on as white colour. The same procedure was performed for the black coloured surface. Finally, the procedures have been repeated several times to accommodate different intensity of ambient light (i.e. whether office lights were turned on or off) and various situations occurring while the mobile robot is moving (readings while the light sensor oscillates during movement of the robot). The training set could be seen in Fig. 3c. Approximately 5,000 readings were gathered for both colours. The learning process was performed in Matlab® environment by implementing Probabilistic Neural Network (PNN). PNN could be perceived as the dual of Generalized Regression Neural Networks for classification.

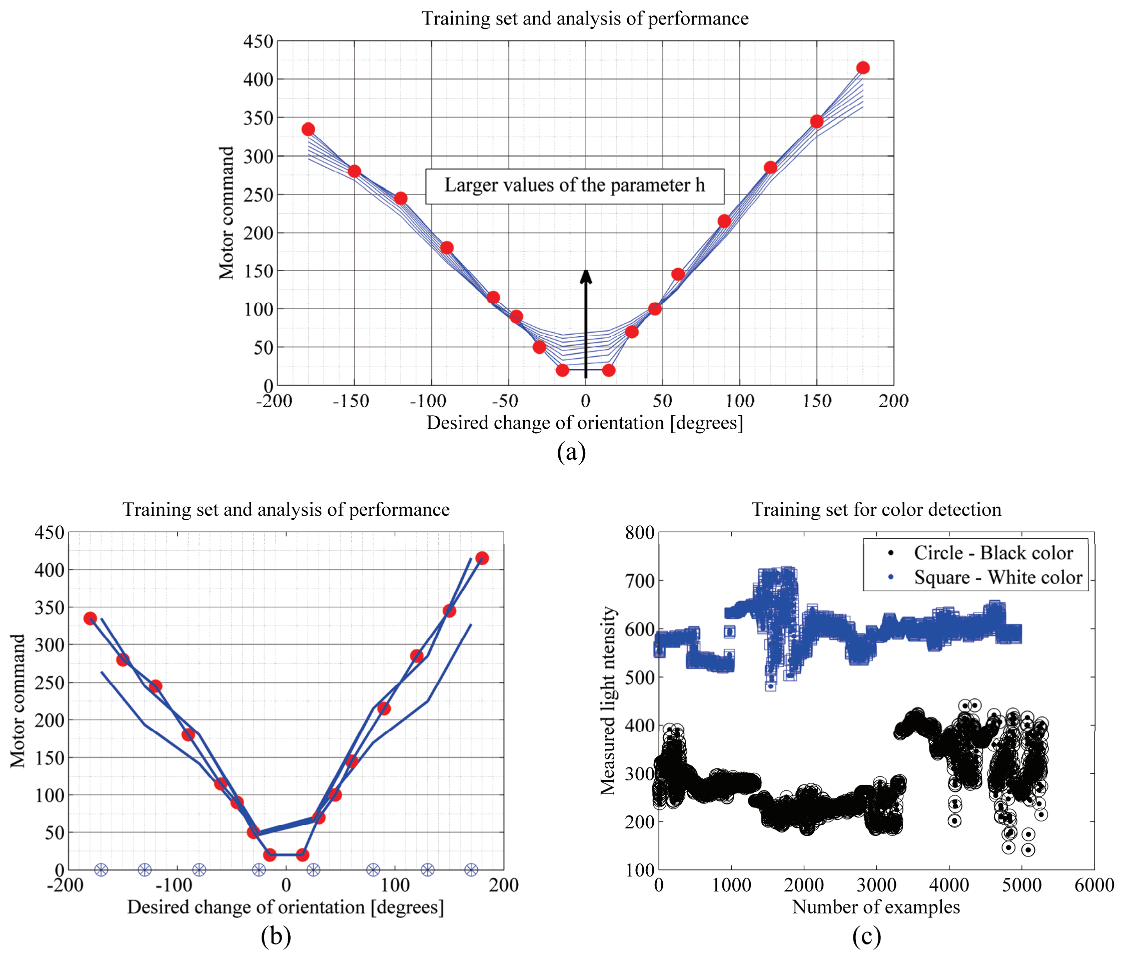
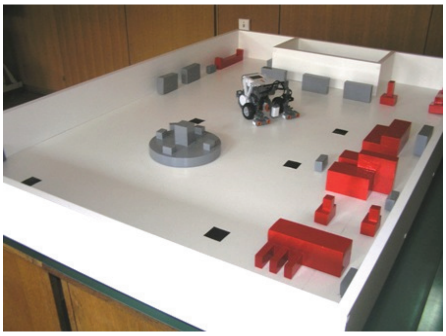


Fig. 3. (a) Training set for steering; (b) Analysis of training results for steering change; (c) Training set for colour detection



(a)

(b)

Fig. 4. Experimental manufacturing environment and the mobile robot

5. EXPERIMENTAL SETUP AND ON-LINE PERFORMANCE IN EXPERIMENTAL ENVIRONMENT

This section describes the mobile robot, its environment, and the specific implementation used throughout the experiments. The experimental model of manufacturing environment was built according to the existing manufacturing enterprise operating near the city where the authors live. Fig. 4 shows the experimental model of the manufacturing environment (a) and the LEGO® Mindstorms NXT mobile robot (b). Configuration consists of two ultrasonic sensors (S_1 and S_2), one light sensor (LS), two motors (Mt_1 and Mt_2) and two encoders (E_1 and E_2) within motors. Ultrasonic sensors were used for obstacle avoidance, two encoders for odometry prediction, and finally the light sensor for detection of landmarks. Software codes for architectural modules have been written in Matlab® environment. Communication with the robot is achieved by using *RWTH Matlab® toolbox* developed at the University of Aachen – Germany. In this way, the laptop serves as the central processing unit. Sending/receiving information is achieved via USB cable.

Figure 5 shows typical situations during one experimental run. The mobile robot is supposed to pick materials from a shearing machine (Fig. 5a), CNC punch press for punching and blanking, auxiliary materials from one of the local warehouses and transport them to the warehouses W1 and W3 for assembling (Fig. 5a-f). The final step of this task is to transport them to the warehouse of final products (Fig. 5f). This task is just one of the many possible tasks that are being executed according to the manufacturing process. Black painted rectangles are features. Although there is no general method that should provide optimal placement of features in the working environment, for purposes of transportation tasks the landmarks have been placed in front of machine tools (Fig. 5a-f. The Matlab® application developed for tracking the robot's performance is given in Fig. 5g. The circle represents the mobile robot. The start position is on the upper left corner. The straight line is the path generated by the path planner module.

Further on we illustrate the performance of the IMHS and overall manufacturing system by simulation experiments. The simulation model is developed using AnyLogic® 6 simulation software – educational version. Graphical presentation of the developed model during animation is given in Fig. 6. For this simulation experiment, the practical inputs of the simulation are shown in Table 2. Simulation results showing workstation utilization are given in Table 3.

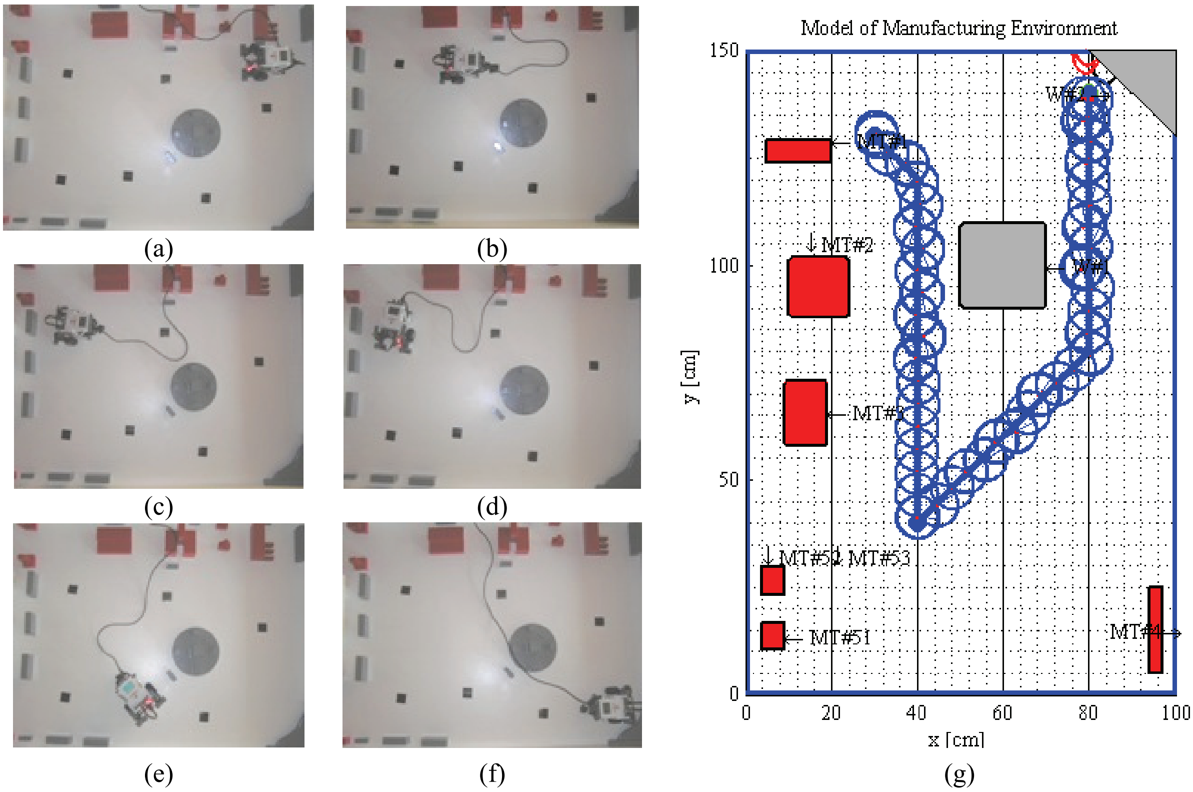


Fig. 5. The mobile robot in the environment while performing the transportation task (a-f).
(g) Matlab® application for tracking the on-line performance

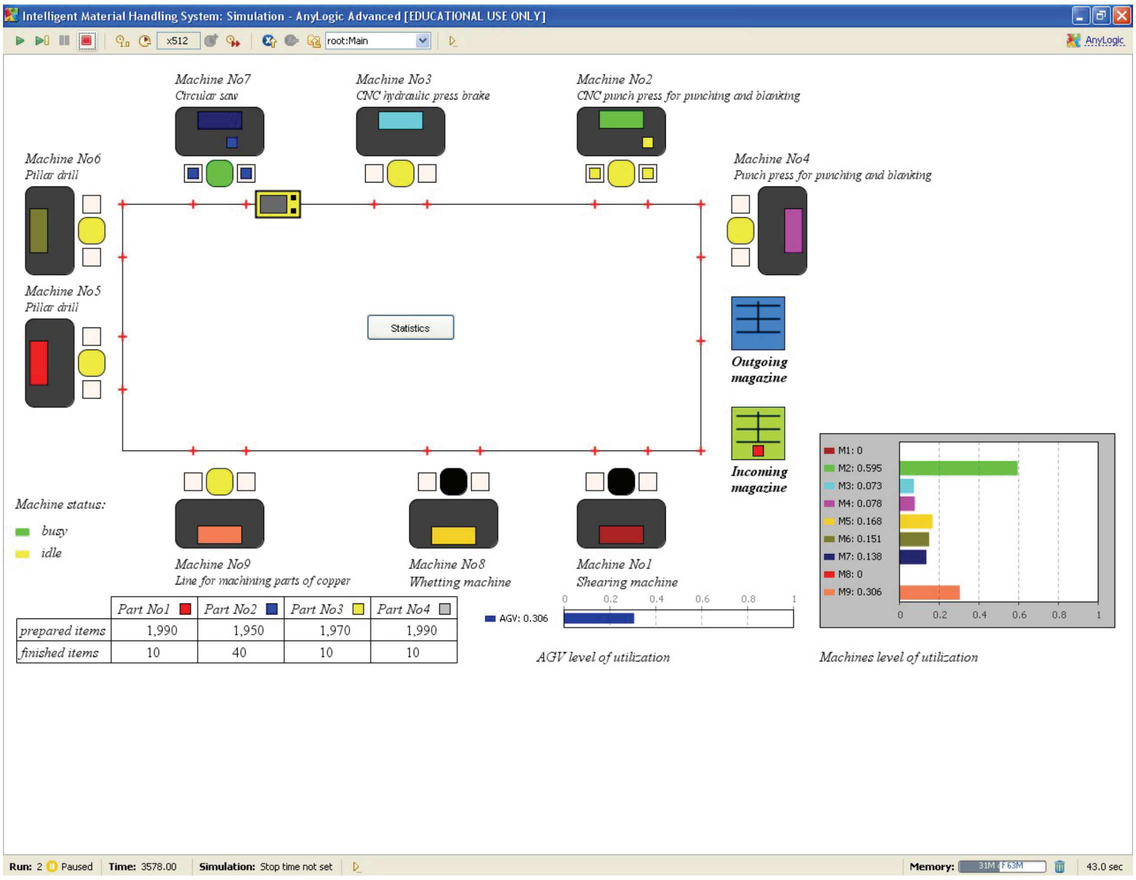


Fig. 6. Graphical presentation of the simulation model during animation

Table 2. Input time for simulation – Operation sequence and operation times (in seconds) for representative parts

Workstation	Transport fuse	Mainbusbar support	Support d800	Busbar 2 L1
M1	-	-	-	-
M2	6+2	-	-	-
M3	2+2	-	4+4+4+2+2	-
M4	2.5	-	32+37+6+3.5+18	-
M5 and M6	-	10+6	-	-
M7	-	12	-	-
M8	-	-	-	-
M9	-	-		2+23+4+7+8+5

Table 3. Simulation results

Workstation #	Utilization of workstation [%]	Average waiting time [s]
1	-	-
2	89.49	35.57
3	61.38	77.88
4	61.42	173.19
5	58.56	127.53
6	52.98	135.22
7	58.62	127.2
8	-	-
9	87.82	127.12

6. CONCLUSION

We have presented the experimental results of the hybrid architecture for AGV navigation and exploitation in a manufacturing environment. We used LEGO® Mindstorms NXT mobile robot and placed it in a laboratory model of existing manufacturing environment. We have shown that the proposed method of receiving and interpreting information can be applied for purposes of autonomous navigation.

One of the most important characteristics of this architecture is its modularity. Individual modules can be changed, reformulated and even redeveloped, not effecting overall performance of the system in the negative sense. If basic blocks of the architecture are upgraded or redeveloped based on faster and more reliable methods, it would not degrade overall performance. For instance, so far we have used EKF for state estimation, but we are currently developing Particle Filter localization [50].

The main idea behind development of this architecture can be summarized in the following sentence: if each of the individual modules performs in an optimal way the entire system will perform better. However, the architecture can be improved. Firstly, the MDF module is focused exclusively on the layout without considering the production process itself. Some ideas for the simultaneous scheduling and AGV routing can be found in [30-33, 61]. It would be interesting to see whether the results and findings reported in [62] could be combined in the framework proposed in this paper to improve the performance. Secondly, the obstacle avoidance is not provided but we have already taken steps in this direction [63]. Thirdly, the state estimation should be extended to the Simultaneous Localization and Mapping (SLAM) [17], which is important for daily exploitation [4]. Fourthly, by implementing a camera we can significantly improve the robot’s performance. Kelly et al. were among the first who implemented vision for AGVs [15]. Advances in visual SLAM [17, 18] provide the basic foundation of this line of future research.

Development of this architecture can be observed from a broader picture as well. Modularity of architecture allows the implementation of various algorithms for basic building blocks (modules for state estimation, path and action planning etc.). This feature has been exploited through classes while teaching Intelligent Manufacturing Systems to master level students at the University of Belgrade-Faculty of Mechanical Engineering (http://cent.mas.bg.ac.rs) [64].

Finally, to evaluate the efficiency of the IMHS and overall manufacturing system, the simulation model is developed. Our simulation results show that the IMHS has advantages over the conventional systems in terms of manufacturing system effectiveness and workstation utilization. Simulation results show good efficiency of the manufacturing system (Table 3). The average utilization of workstations is 67.20%, and utilization of IMHS is 42%. Effectiveness of the manufacturing system is 62 products per hour. Therefore, the IMHS can speed up the manufacturing process, lower the inventory cost, and have the capability of fast response to the customer demands. Due to product proliferation, this potential advantage is important for the metalworking industry. Although the obtained results showed usefulness further research is needed to achieve the demanded level of reliability imposed by the complexity of the transportation task itself.

Acknowledgements**:** This work was supported by the Serbian Government - the Ministry of Education and Science-Project title: An innovative, ecologically based approach to the implementation of intelligent manufacturing systems for the production of sheet metal parts (2011-2014) under grant TR35004.

The authors would like to thank their partners in SAGA Company (www.saga.rs) as well, for their help and much needed support achieved by donating the Hewlett-Packard notebook PC HP Compaq 6820s which is being used for experiments. Their help is gratefully acknowledged.

NOMENCLATURE

x_t	state vector at time instant t
x_{t-1}	state vector at time instant $t-1$
u_t	the control vector at time t
F_t	a square matrix of size n x n (n is the dimension of state vector x_t)
B_t	matrix of size m x n (m is the dimension of the control vector u_t)
w_t	the zero mean uncorrelated motion noise with the Gaussian distribution
Q_t	covariance matrix of w_t
z_t	measurement vector
H_t	a matrix of size k x n
v_t	the measurement noise with the Gaussian distribution with zero mean uncorrelated noise
R_t	covariance matrix v_t
$h(x_t)$	the nonlinear function that defines the measurement model
G_t	Jacobian of motion model with respect to the state vector x_{t-1}
V_t	Jacobian of motion model with respect to the control vector u_t
H_t	measurement Jacobian with respect to state vector x_{t-1}
$f(z_t)$	feature vector
x_t^n, y_t^n	x and y coordinates of feature n defined in global frame
$g(x_{t-1}, u_t)$	the nonlinear function that defines the state transition model
s_t^n	signature of feature
A*	search algorithm
$f(n)$	cost function
$g(n)$	path-cost function which represents cost of going from node i to node j
$h(n)$	heuristic that determines the distance from node j to the goal
y_i	nonlinear regression function
x_i	the input vector ($x \in \Re^m$)
y_i	the observable ($y \in \Re$)
$f(x_i)$	regression function
$K(\cdot)$	Kernel function
h	parameter controlling the size of kernel $K(\cdot)$
S_1, S_2	ultrasonic sensors
Mt_1, Mt_2	motors
E_1, E_2	encoders within motors
W1, W2, W3, W4 and W5	warehouses
M1, M2, M3, M4, M5, M6, M7, M8, M9	machine symbols

FMS	Flexible Manufacturing System	EKF	Extended Kalman Filter
CNC	Com puter Numerically Controlled	ICAPP	Intelligent Computer Production Planning
MHS	Material Handling System	SCNN	Steering Control Neural Network
AGV	Automated Guided Vehicle	GRNN	Generalized Regression Neural Network
IMHS	Autelligent Material Handling System	RBF	Radial Basis Functions
MDF	Mission Data File	PNN	Probabilistic Neural Network
KF	Kalman Filter	LS	Light Sensor
δ	the Kronecker delta function	ε_i	white noise with zero mean and variance σ^2
μ_t	expected value of the robot state	$\sum_t$	covariance matrix at time t

REFERENCES

1. Babic, B. (1999). Axiomatic design of flexible manufacturing systems. *International Journal of Production Research*, Vol. 37, No. 5, pp. 1159 – 1173.
2. Beamon, B. M. (1998). Performance, reliability, and performability of material handling systems. *International Journal of Production Research*, Vol. 36, No. 2, pp. 337–393.
3. Jawahar, N., Aravindan, P., Ponnambalam, S. G. & Suresh, P. K. (1998). AGV schedule integrated with production in flexible manufacturing systems. *International Journal of Advanced Manufacturing Technology*, Vol. 14, No. 6, pp. 428–440.
4. Martinez-Barbera, H. & Herrero-Perez, D., (2010). Autonomous navigation of an automated guided vehicle in industrial environments. *Robotics and Computer-Integrated Manufacturing*, Vol. 26, pp. 296–311.
5. Groover, M. P., (2001). *Automation, production systems, and computer-integrated manufacturing*. (2nd edition), New Jersey, Prentice Hall.
6. Egbelu, P. J. & Tanchoco, J. M. A. (1986). Potentials for bi-directional guide-path for automated guided vehicle based systems. *International Journal of Production Research*, Vol. 24, No. 5, pp. 1075–1097.
7. Afentakis, P. (1989). A loop layout design problem for flexible manufacturing systems. *International Journal of Flexible Manufacturing Systems*, Vol. 1, No. 2, pp. 175–196.
8. Bozer, Y. A. & Srinivasan, M. M., (1989). Tandem configurations for AGV systems offer simplicity and flexibility. *Industrial Engineering.* Vol. 21, No. 2, pp. 23–27.
9. Bozer, Y. A. & Srinivasan, M. M., (1991). Tandem configurations for automated guided vehicle systems and the analysis of single vehicle loops. *IIE Transactions.* Vol. 23, No. 1, pp. 72–82.
10. Bozer, Y. A. & Srinivasan, M. M. (1992). Tandem AGV systems: A partitioning algorithm and performance comparison with conventional AGV systems. *European Journal of Operational Research,* Vol. 63, No. 2, pp. 173–191.
11. Tanchoco, J. M. A. & Sinriech, D., (1992). OSL—optimal singleloop guide paths for AGVS. *International Journal of Production Research.* Vol. 30, No. 3, pp. 665–681.
12. Kim, C. W. & Tanchoco, J. M. A., (1993). Operational control of a bidirectional automated guided vehicle system. *International Journal of Production Research*, Vol. 31, No. 9, pp. 2123–2138.
13. Rajotia, S., Shanker, K. & Batra, J. L. (1998). An heuristic for configuring a mixed uni/bidirectional flow path for an AGV system. *International Journal of Production Research*, Vol. 36, No. 7, pp. 1779–1799.
14. Yu, W. & Egbelu, P. J. (2001). Design of a variable path tandem layout for automated guided vehicle systems. *Journal of Manufacturing Systems.* Vol. 20, No.5, pp. 305–319.
15. Kelly, A., Nagy, B., Stager, D. & Unnikrishnan, R. (2007). An infrastructure-free automated guided vehicle based on computer vision. *IEEE Robotics and Automation Magazine*, Vol. 14, No. 3, pp.24–34.
16. Martinez-Barbera, H. & Herrero-Perez, D., (2010). Development of a flexible AGV for flexible manufacturing systems. *Industrial Robot: An International Journal*, Vol. 37, No.5, pp. 459–468.
17. Civera, J., Grasa, O. G. & Davison, A. J. (2010). 1-Point RANSAC for EKF Filtering. Application to Real-Time Structure from Motion and Visual Odometry. *Journal of Field Robotics - Visual Mapping and Navigation Outdoors*. Vol. 27, No. 5, pp. 609–631.

18. Scaramuzza, D., Fraundorfer, F. & Pollefeys, M. (2010). Closing the loop in appearance-guided omnidirectional visual odometry by using vocabulary trees. *Robotics and Autonomous Systems*, Vol. 58, No. 6, pp. 820-827.
19. Nister, D., Naroditsky, O. & Bergen, J. (2004). Visual odometry. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 652-659.
20. Arkin, R. C. & Murphy, R. R. (1990). Autonomous navigation in a manufacturing environment. *IEEE Transactions on Robotics and Automation*, Vol. 6, No. 4, pp. 445-454.
21. Zhou, J., Deng, J. X., Huang, P. L., Liu, Z. Q. & Ai, X. (2006). Integrated analysis method: visual modelling, simulation, diagnosis and reduction for bottleneck processes of production lines*, Iranian Journal of Science & Technology, Transaction B: Engineering*, Vol. 30. No. B3, pp. 363-375.
22. Le-Anh, T. & De Koster, M. B. M., (2006). A review of design and control of automated guided vehicle systems. *European Journal of Operational Research,* Vol. 171, No.1, pp.1–23.
23. Vis, I. F. A. (2006). Survey of research in the design and control of automated guided vehicle systems. *European Journal of Operational Research,* Vol. 170, No. 3, pp. 677–709.
24. Durrant-Whyte, H. F. (1996). An autonomous guided vehicle for cargo handling applications. *International Journal of Robotics Research*, Vol. 15, No.5, pp. 407–440.
25. Durrant-Whyte, H. F., Pagac, D., Rogers, B., Stevens, M. & Nelmes, G. (2007). An autonomous straddle carrier for movement of shipping containers. *IEEE Robotics and Automation Magazine*, Vol. 14, No. 3, pp. 14–24.
26. Takahashi, M., Suzuki, T., Shitamoto, H., Moriguchi, T. & Yoshida, K., (2010). Developing a mobile robot for transport applications in the hospital domain. *Robotics and Autonomous Systems*, Vol. 58, No. 7, pp. 889-899.
27. Kelly, A. & Nagy, B. (2003). Reactive nonholonomic trajectory generation via parametric optimal control. *International Journal of Robotics Research*, Vol. 22, No. 7–8, pp. 583–601
28. Kelly, A. (2000). Mobile robot localization from large scale appearance mosaics. *International Journal of Robotics Research*, Vol. 19, No. 11, pp. 1104–1125.
29. Qiu, L., Hsu, W., Huang, S. & Wang, H. (2002). Scheduling and routing algorithms for AGVs: a survey. *International Journal of Production Research*, Vol. 40, No. 3, pp.745–760.
30. Lacomme, P., Larabi, M. & Tchernev, N. (2010). Job-shop based framework for simultaneous scheduling of machines and automated guided vehicles. *International Journal of Production Economics*, doi:10.1016/j.ijpe.2010.07.012
31. Nishi, T., Hiranaka, Y. & Grossmann, I. E. (2011). A bilevel decomposition algorithm for simultaneous production scheduling and conflict-free routing for automated guided vehicles. *Computers & Operations Research*, Vol. 38, No. 5, pp. 876–888.
32. Corréa, A. I., Langevin, A. & Rousseau, L-Martin, (2007). Scheduling and routing of automated guided vehicles: A hybrid approach. *Computers & Operations Research*, Vol. 34, No. 6, pp.1688–1707.
33. Angeloudis, P. & Bell, M. (2010). An uncertainty-aware AGV assignment algorithm for automated container terminals. *Transportation Research Part E*., Vol. 46, No. 3, pp. 354–366.
34. Berman, S., Schechtman, E. & Edan, Y. (2009). Evaluation of automatic guided vehicle systems. *Robotics and Computer-Integrated Manufacturing*, Vol. 25, No. 3, pp. 522–528.
35. Berman, S. & Edan, Y. (2002). Decentralized autonomous AGV system for material handling. *International Journal of Production Research*, Vol. 40, No. 15, pp. 3995–4006.
36. Berman, S., Edan, Y. & Jamshidi, M. (2003). Decentralized autonomous automatic guided vehicles in material handling. *IEEE Transactions on Robotics and Automation*, Vol. 19, No. 4, pp. 743–749.
37. Wang, Z. D., Nakano, E. & Takahashi, T. (2003). Solving function distribution and behavior design problem for cooperative object handling by multiple mobile robots. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, Vol. 33, No.5, pp. 537 – 549.
38. Arkin, R., (1998). *Behavior-Based Robotics*, Cambridge MA: MIT Press.

39. Gat, E., (1998). Three-layered architectures, in: D. Kortenkamp, R. Bonaso and R. Murphy (Ed.), *AI-based mobile robots: Case studies of successful robot systems*, Cambridge, Massachusetts, pp. 195-210.
40. Na, Y. K. & Oh, S. Y. (2003). 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.
41. Vuković, N. & Miljković, Z. (2009). New hybrid control architecture for intelligent mobile robot navigation in a manufacturing environment. *FME Transactions*, Vol. 37, No.1, pp. 9-18.
42. Fukuda, T. & Arakawa, T. (1998). Intelligent systems: Robotics versus mechatronics. *Annual Reviews in Control*, Vol. 22, pp. 13-22.
43. Thrun, S., Burgard, W. & Fox, D. (2005). *Probabilistic robotics*. Cambridge MA: MIT Press.
44. Leonard, J. J. & Durant-Whyte, H. F. (1991). Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 3, pp. 376-382.
45. Betke, M. & Gurvits, L. (1997). Mobile robot localization using landmarks. *IEEE Transactions on Robotics and Automation*, Vol. 13, No.2, pp. 251-263.
46. Saffiotti, A. (1997). The uses of fuzzy logic in autonomous robot navigation. *Soft Computing*, Vol. 1, No. 4, pp. 180-197.
47. Nourbakhsh, R. (1998). Dervish: An office-navigation robot. *In*: D. Kortenkamp, R. Bonaso and R. Murphy, eds. *Artificial Intelligence and Mobile Robots, Case Studies of Successful Robot Systems*. Cambridge MA: MIT Press, pp.73-90.
48. Townsend, N. W. & Tarassenko, L. (1999). Neural networks for mobile robot localisation using infra-red range sensing. *Journal Neural Computing & Applications,* Vol. 8, No.2, pp. 114-134.
49. Hu, H. & Gu, D. (2000). Landmark–based Navigation of Industrial Mobile Robots. *Industrial Robot. An International Journal*, Vol. 27, No. 6, pp. 458 – 467.
50. Fox, D., Thrun, S., Dellaert, F. & Burgard, W. (2001). Particle filters for mobile robot localization. *In*: A. Doucet, N. de Freitas and N. Gordon, eds. *Sequential Monte Carlo Methods in Practice*, Berlin Heidelberg: Springer Verlag, pp. 499-516.
51. Moreno, L., Garrido, S. & Munoz, M. L. (2006). Evolutionary filter for robust mobile robot global localization. *Robotics and Autonomous Systems*, Vol. 54, No. 7, pp. 590-600.
52. Hui, N. B. & Pratihar, D. K. (2008). Soft computing-based navigation schemes for a real wheeled robot moving among static obstacles. *Journal of Intelligent and Robotic Systems*, Vol. 51, No. 3, pp. 333–368.
53. Yang F., Wang, Z., Lauria, S., & Liu, X. (2009). Mobile robot localization using robust extended H_∞ filtering. *Proceedings of Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, Vol. 223, No. I8, pp. 1067-1080.
54. Bar-Shalom, Y., Li, X. R. & Kirubarajan, T. (2001). *Estimation with applications to tracking and navigation-theory, algorithms and software.* New York: John Wiley & Sons.
55. Simon, D. (2006). *Optimal state estimation: Kalman, H-infinity, and nonlinear approaches*. New York: John Wiley & Sons.
56. Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME-Journal of Basic Engineering*, Vol. 82 (Series D), pp. 35-45.
57. Stentz, A. (1996). Map-based strategies for Robot navigation in unknown environments. AAAI spring symposium on planning with incomplete information robot problems. Menlo Park, CA: AAAI Press; pp. 110–16.
58. Stentz, A. (1995). The focused D* algorithm for real-time replanning. In: International Joint Conference on Artificial Intelligence, Taiwan, pp. 1652–9.
59. Haykin, S. (1998). *Neural networks: A comprehensive foundation (2nd edition)*. Upper Saddle River, New Jersey: Prentice Hall.
60. Dreyfus, G., ed. (2005). *Neural networks – methodology and applications*. Berlin Heidelberg: Springer-Verlag.

61. Furusho T., Nishi T. & Konishi, M. (2008). Distributed optimization method for simultaneous production scheduling and transportation routing in semiconductor fabrication bays. *International Journal of Innovative Computing, Information and Control*. Vol. 4, No. 3, pp. 559-575.
62. Debevec, M. & Herakovic, N. (2010). Management of resources in small and Medium-sized production enterprises, *Iranian Journal of Science & Technology, Transaction B: Engineering,* Vol. 34, No. B5, pp. 509-520.
63. Vuković, N., Miljković, Z., Babić, B. & Čović, N. (2011). Towards implementation of intelligent mobile robots in a manufacturing environment. *Proceedings of the 4th International Conference on Manufacturing Engineering (ICMEN 2011)*, Thessaloniki, Greece, pp. 367-376.
64. Petrović M., Lukić N., Vuković N., & Miljković, Z. (2010). Mobile robot for material transport in intelligent manufacturing system-education and development (in Serbian). *Proceedings of the 36th JUPITER Conference*, Belgrade, Republic of Serbia, 3.85-3.90.