Intelligent mobile manipulator navigation using adaptive neuro-fuzzy systems

Jean Bosco Mbede a,b,c,*, Pierre Ele b, Chantal-Marguerite Mveh-Abia b, Youssoufï Toure c, Volker Graefe d, Shugen Ma a

a Faculty of Engineering, Ibaraki University, Hitachi, Ibaraki 316-8511, Japan
b Ecole Nationale Supérieure Polytechnique, Université de Yaoundé I, BP 8390 Yaoundé, Cameroun
c Laboratoire de Vision et Robotique, Université d’Orléans, 18020 Bourges, France
d Faculty of Aerospace Engineering, The University of the Armed Forces of Munich, 85577 Neubiberg, Germany

Accepted 27 September 2004

Abstract

The work presented in this paper deals with the problem of autonomous and intelligent navigation of mobile manipulator, where the unavailability of a complete mathematical model of robot systems and uncertainties of sensor data make the used of approximate reasoning to the design of autonomous motion control very attractive.

A modular fuzzy navigation method in changing and dynamic unstructured environments has been developed. For a manipulator arm, we apply the robust adaptive fuzzy reactive motion planning developed in [J.B. Mbede, X. Huang, M. Wang, Robust neuro-fuzzy sensor-based motion control among dynamic obstacles for robot manipu-
lators, IEEE Transactions on Fuzzy Systems 11 (2) (2003) 249–261]. But for the vehicle platform, we combine the advantages of probabilistic roadmap as global planner and fuzzy reactive based on idea of elastic band. This fuzzy local planner based on a computational efficient processing scheme maintains a permanent flexible path between two nodes in network generated by a probabilistic roadmap approach. In order to consider the compatibility of stabilization, mobilization and manipulation, we add the input of system stability in vehicle fuzzy navigation so that the mobile manipulator can avoid stably unknown and/or dynamic obstacles. The purpose of an integration of robust controller and modified Elman neural network (MENN) is to deal with uncertainties, which can be translated in the output membership functions of fuzzy systems.

© 2004 Elsevier Inc. All rights reserved.

Keywords: Autonomous mobile manipulator; Fuzzy reactive navigation; Robust neuro-fuzzy controllers

1. Introduction

Mobile manipulator is nowadays a widespread term to refer to robots built from a robotic arm mounted on a mobile platform. Such systems combine the advantages of mobile platforms and robotic arms and reduce their drawbacks. For instance, the mobile platform extends the arm workspace, whereas an arm offers much operational functionality. Applications for such systems abound in mining, construction, forestry, planetary exploration, and the military. Although appeared very early in robotics history [13], a host of issues related to mobile manipulators have mainly been studied in recent years [3,23,24]. These include dynamic and static stability, force development and application, control in the presence of base compliance, dynamic coupling issues [12,17]. Particularly, motion planning has generally been treated within the framework of optimal control. The complete trajectory is preplanned before execution. On these issues several papers have appeared in literature: Desai and Kumar [4] have formulated the problem as an optimal control problem. Pin et al. [15] performed local optimisation at velocity level. Huang et al. [8] decoupled the motion of the vehicle from that of the manipulator and optimised each one using different criteria. Recently, Tanner et al. [21] proposed a closed loop motion-planning scheme that achieves convergence and obstacle avoidance based on the potential field method. Because generally the precedent method might fail to achieve the task by getting trapped in local minima, Brock and Khatib [2] combine the advantages of reactive controllers (fastness) and the advantages of planners (find a solution whenever one exists) by an elastic band or elastic strip formulation. Siméon et al. [19] provided a probabilistic scheme in order to compute the visibility roadmaps. In most of the previous works, the robot workspaces are highly engineered and completely determined at design time.
The efficacy of motion control algorithms need to be validated in face of uncertainty and imprecision that are inherent of real-time navigation and motion control based on poor and noisy sensory data.

The fuzzy control is credited with being an adequate methodology for designing robust controllers that are able to deliver a satisfactory performance in face of large amounts of noise in the input and of variability in the parameters. This methodology attempts to build the fuzzy rules and the (inputs and outputs) membership functions based on expert knowledge. Significant amounts of research and development have been performed through the observation of the actions of a human operator controlling the mobile robots [7,9,18,22], and in the areas of autonomous motion control of manipulator [1,6,10,11]. But very little has been reported on the topic of coupling these two functions for the purpose of developing an autonomous navigation of mobile manipulator in dynamic unstructured environments. The uncertainties associated with changing and dynamic unstructured environments cannot be handled by using only precise and crisp membership functions of fuzzy sets. Erden et al. [5] identified the base, elbow and shoulder as three different agents to control a mobile manipulator using fuzzy control. They explored the use of genetic algorithms to turn the membership functions of fuzzy controllers.

In this paper, we attempt to develop an on-board intelligent motion controller based on fuzzy obstacle avoidance scheme, that enable the mobile manipulator to navigate autonomously to specified goals, in presence of uncertainties, while not exposing the robot to any undue risk. We couple the fuzzy reactive navigation algorithm of manipulator arm developed in [10] with autonomous navigation of mobile platform. The autonomous navigation of mobile platform combines the advantages of probabilistic roadmap as global planner and the advantages of fuzzy reactive planner based on idea of elastic band. The proposed intelligent local planners have many advantages: adaptation to fast changes in environment and real-time sensor-based path planning application. These attributes make them suitable for on-line dynamic obstacle avoidance. Furthermore, it is necessary to consider the stability problem (overturn prevention), because the mobile manipulator might tip over when the vehicle is moving at a high speed without taking into the optimal configuration of the arm or in environment with disturbance, and when the arm is exerting large forces to move heavy payload. The purpose of the system stability input of fuzzy navigation of vehicle control is to maintain the system stable during its motion.

Unlike fuzzy-genetic motion planning [5], robust fuzzy obstacle avoidance should be capable of dealing with structured and unstructured uncertainties in the output of intelligent systems. To accomplish this task, we proposed an integration of robust terms and modified Elman neural network (MENN) to approximate such uncertainties. We preferred this MENN to a multilayer neural network, because there is not yet a well-defined procedure for the selection of optimum multilayer neural network architecture for a given problem. The
common technique is to try a few architectures that give the required approximation while keeping the complexity of the network at a low level. The MENN is a recurrent neural network used by Pham and Liu [14] to identify the second order nonlinear dynamic system. The proposed MENN controller differs from Pham and Liu’s learning techniques in the sense that the adaptation algorithm is computed on-line using the operational data of actual robot movements based on the trajectory planning given by fuzzy navigation. To keep the weights of the MENN from growing without bounds, we use a robust term in control law. In real-time world systems, sensor-based motion control becomes essential to deal with model uncertainties and unexpected obstacles.

The rest of this paper is organised as follows. In Section 2, we introduce a modular model of mobile manipulator containing actuator dynamic. Before presenting the design of robust neuro-fuzzy controller in Section 4, we first describe the navigation techniques based on combination of two fuzzy local planners and global probabilistic roadmap approach, and derive the coordination motion control algorithm in Section 3. Section 5 presents our simulation results. Finally, Section 6 gives some concluding remarks.

2. Modular model containing actuator dynamics

The equations of motion containing the actuator dynamics for a mobile manipulator, which the skeleton model is shown in Fig. 1, are given by

\[
M_a(q_a)\ddot{q}_a + C_a(q_a, \dot{q}_a, \dot{q}_v) + G_a(q_a, q_v) + F_a(\dot{q}_a, \dot{q}_v) = \tau_a - R_a(q_a, q_v)\ddot{q}_v, \\
M_v(q_a, q_v)\ddot{q}_v + C_v(q_a, q_v, \dot{q}_a, \dot{q}_v) + G_v(q_a, q_v) + F(\dot{q}_a, \dot{q}_v) = E_v\tau_v - A^T\lambda - R_v(q_a, q_v)\ddot{q}_a,
\]

(1)

\[
\tau_a = N_aK_a^T\tilde{I}_a, \\
\]

(2)

\[
r_a\dot{I}_a + L_a\dot{\tilde{I}}_a + N_aK_a^e\ddot{q}_a = u_a,
\]

(3)

\[
\dot{q}_v = \begin{bmatrix}
\cos \theta_v & -d \sin \theta_v \\
\sin \theta_v & d \cos \theta_v \\
0 & 1
\end{bmatrix}
\begin{bmatrix}
\frac{\pi r_c N_v K_v^m (u_r + u_l)}{2\pi r_c N_v K_v^m (u_r - u_l)} \\
\end{bmatrix},
\]

(4)

where \(q_a, q_v, \dot{q}_a, \dot{q}_v, \ddot{q}_a, \ddot{q}_v\) are the coordinates, velocities and accelerations of arm robot and its vehicle, respectively, \(M_a(q_a)\) is the inertia matrix of manipulator, \(C_a(q_a, \dot{q}_a, \dot{q}_v)\) represents coriolis and centrifugal terms of manipulator, and coriolis and centrifugal terms caused by the angular motion of the platform, \(\tau_a\) is the input torque/force for the manipulator, and \(R_a(q_a, q_v)\) is the inertia matrix
which represents the effect of the vehicle dynamics on the arm. The generalized joint torque equation (3) contains the torque constant of the \( j \)th motor \( K_a \) and the gear ratio of the \( j \)th joint \( N_a \). The voltage equation (4) of the armature circuit of the \( j \)th motor of the robot arm contains the resistance of armature circuit \( r_a \), \( L_a \) the inductance of armature current, the armature current \( I_a \), the voltage constant of the motor \( K_e_a \), and the armature voltage \( u_a \). The mobile platform’s motion equations (2) and (5) contain the voltage constant of the motor \( K_m \), the armature voltage of the right wheel \( u_r \), the armature voltage of the left wheel \( u_l \), the number of rotation of both motors \( N_v \), the radius of each driven wheel \( r_v \), the distance between the driven wheels and the axis of symmetry \( b \), the angular velocity of wheel \( \dot{\theta}_v \), where \( d \) is the distance from \( P_0 \) (the intersection of the axis of symmetry with the driving wheel axis) to \( P_c \) (the center of mass of the platform). \( M_v(q_a, q_v) \) represents the mass inertia matrix of platform and the inertia term due to the presence of the manipulator, \( C_v(q_a, q_v, \dot{q}_a, \dot{q}_v) \) is the velocity dependent terms of the platform, and the coriolis and centrifugal terms due to the presence of the manipulator, \( \tau_v \) is the input
torque/force to the vehicle, $E_v$ is a constant matrix, $A$ is a Jacobian matrix associated with the contact surface geometry, $\lambda$ denotes the vector of Lagrange multiplier corresponding to the kinematic constraints, and $R_v(q_a, q_v)$ represents the inertia matrix which reflects the dynamic effect of the arm motion on the vehicle.

3. Mobile manipulator navigation

For mobile manipulator path planning, we describe a new paradigm to the learning motion planning approach, which combines a global roadmap with fuzzy local planners. In the learning phase, which is closely related to the works done by [20], a probabilistic roadmap is built up by repeatedly generating random free configurations and trying to connect these to other configurations by some simple motion planning algorithm. The network thus formed is stored in a graph $G$. After the learning is done, a query consists of trying to connect the given start and goal configurations to some nodes (in the same connected component) of $G$, with paths, which are feasible for the robot. Next, local planners can transform the path between these nodes in $G$ into a feasible path. Two local planners have been implemented for our mechanical system. The first one is applied to arm manipulator path. This local planner is based on fuzzy rules for idea of artificial potential fields using analytic harmonic functions. The second local planner takes into account the nonholonomic constraints of mobile platform in Reeds and Shepp metric system [16]. This last local planner is also based on fuzzy rules but for idea of the concept of elastic band in Reeds and Shepp metric system, which consists in maintaining a permanent flexible path between two nodes in $G$.

3.1. Reactive fuzzy navigation of arm manipulator

In this subsection, we realize a collision avoidance based on the local intelligent dynamic motion-planning algorithm presented in Fig. 2 where a current position of robot arm $X_A = h(q_a)$ is the transformation from joint coordinates to Cartesian coordinates, and $X_G$ is a set of points given by operator (goals) and/or imposed by system stability.

It is important to note that the case where $d_a > d_{\text{min}}$, is extremely rare because it is a moment when the obstacles suddenly surround the robot. The Fuzzy decision is described in [10].

3.2. Reactive fuzzy navigation of mobile platform

The mobile platform reactive navigation is based on local intelligent dynamic motion planning algorithm presented in Fig. 3. The fuzzy system, which
Consider a set of points $X_0 = \{X_{i,0}, X_{i,1}, \ldots, X_{i,n}\}$

Give a start position $X_{o,i}$ and the minimum approach distance $d_a$

Read the first point
$X_0 = X_{o,i}$

Read the start position
$X_A = X_{o,i}$

if $X_A = X_0$

no

Read sensor measures $d_i$

Compute $d_{\min} = \min(d_i)$

if $d_i > d_{\min}$

yes

Compute obstacle segments

Compute $d = d_{\min} - d_i$ and $e = X_{o,i} - X_A$

Fuzzy decision: Arm motion

Compute $X_A = h(q_i)$

if $X_0 - X_A \neq 0$

no

yes

One point is reached

if $d_i = d_{\min}$

yes

Unknown obstacle obscures the point

if Does it move in direction of robot?

no

yes

Unsuccessful planning (end)

Collision (end)

Successful planning (end)

if Is there a moving obstacle?

no

yes

Stop

Fig. 2. Arm manipulator local intelligent dynamic motion planning algorithm.
makes decision about the vehicle motion, has three types of input variables: sensor data represented by the distance $d_v$ that is the difference between the minimum distance $d_{\text{min}}$ of sensor data and the given minimum approach distance $d_j$.

makes decision about the vehicle motion, has three types of input variables: sensor data represented by the distance $d_v$ that is the difference between the minimum distance $d_{\text{min}}$ of sensor data and the given minimum approach distance $d_j$.

Fig. 3. Vehicle local intelligent dynamic motion planning algorithm.
distance \( d_v \). The difference \( e_v \) between the present localization \( q_v \) of vehicle and the position \( q_{v_i} \) of next node \( n_i \) is the second input of our fuzzy reactive navigation. The last input considers the system stability, mobilization, and manipulation. The force \( F_v \) that approximate the virtual spring forces attached to control points on consecutive configurations of the robot along the elastic strip, and the correction position \( X_{Gi} \) of arm manipulator that maintain the system stable during the vehicle motion are the both output variables of the vehicle robust fuzzy controller.

3.2.1. Fuzzification

The fuzzification module transforms numerical variables in fuzzy sets, which can be manipulated by the controller.

The distance \( d_v \) is partitioned into six fuzzy sets: far left (FL)/far back (FB), medium left (ML)/medium back (MB), close left (CL)/close back (CB), close right (CR)/close front (CF), medium right (MR)/medium front (MF), and far right (FR)/far front (FF). When the obstacle is in the left or back of the vehicle the distance \( d_v \) is negative, and when it is in right or front of the mobile platform, the distance \( d_v \) is positive. Its fuzzy membership functions are asymmetric and shown in Fig. 4 where \( \varphi \) is the maximum ultrasonic range.

The difference \( e_v \) is partitioned into five fuzzy sets: big negative (BN), small negative (SN), zero (Z), small positive (SP), and big positive (BP). Its fuzzy membership functions are symmetric and shown in Fig. 5 where \( \chi \) denotes the maximum distance between two nodes.

The arm position \( q_s \) for the system stability during the vehicle motion is partitioned into three fuzzy sets: left (LF)/back (BC), medium left (ML)/medium back (MB), and right (RG)/front (FR). Its fuzzy membership functions are symmetric and shown in Fig. 6 where \( \vartheta \) is the bound of stability space of arm manipulator during vehicle motion.

3.2.2. Rule base

The rule base is generalized as follows: \( R^i \): If \( e_v(k) \) is \( \mu^i_l(e_v(k)) \) and...and \( e_v(k-n+1) \) is \( \mu^i_n(e_v(k-n+1)) \) and \( d_v(k) \) is and...and \( d_v(k-m+1) \) is...
The outputs of fuzzy rule base are virtual spring forces $F_v$ and stability position of manipulator arm $X_G$. The torque output $F_v$ is partitioned into thirteen fuzzy sets: forward left very big (FLVB)/backward left very big (BLVB), forward left big (FLB)/backward left big (BLB), forward left small (FLS)/backward left small (BLS), forward left very small (FLVS)/backward left very small (BLVS), forward right very big (FRVB)/backward right very big (BRVB), forward right big (FRB)/backward right big (BRB), forward right small (FRS)/backward right small (BRS), forward right very small (FRVS)/backward right very small (BRVS), forward strait very big (FSVB)/backward strait very big (BSVB), forward strait big (FSB)/backward strait big (BSB), forward strait small (FSS)/backward strait small (BSS), forward strait very small (FSVS)/backward strait very small (BSVS), and zero (Z). The stability position $X_G$ is partitioned into three fuzzy sets: left (LF), zero (Z), and right (RG).

For example, $e_v$ is BP, $d_v$ is FF, and $q_s$ is MR then $F_v$ is FSVB and $X_G$ is Z.

### 3.2.3. Defuzzification

Two output fuzzy sets result from the inference process, one for the force $F_v$, and the other for the correction position $X_{Gi}$ of arm manipulator. The defuzzification module obtains, from these sets, numerical values. It was chosen the
center of gravity (COG) method because it takes into account, better than any other method, the distribution of the resultant fuzzy set. The COG method, applied to fuzzy singletons, simplifies the computation of inference mechanism and reduces processing time without degradation of the defuzzified value. This one is obtained from:

$$u = \frac{\sum a_i s_i}{\sum a_i},$$  \hspace{1cm} (6)

where \(a_i\) is the degree of activation of the \(i\)th rule and \(s_i\) is the output singleton.

### 3.3. Cooperative motion planning

Considering the compatibility of high velocity of the vehicle, manipulation of the manipulator and stabilization motion of all the system, it is necessary to coordinate the two local planners. Then, based on the assurance of system stability, the mobile manipulator should execute tasks with an optimal configuration. The vehicle motion mainly affects system stability: for example, the robot falls over easily if the velocity or acceleration of the vehicle is high. Since the range of motion of the manipulator will not be large given an end-point trajectory, the manipulator has only limited ability to aid stability. Therefore; in order to ensure system stability, it is effective to plan vehicle local motion first; then plan manipulator local motion. However, if considering only vehicle motion and system stability while planning vehicle motion, the manipulator

![Fig. 7. Algorithm of cooperative motion planning.](image-url)
configuration will not be suitable to execute end-point tasks. Conversely, if considering only optimal configuration for executing end-point tasks while planning manipulator motion, the velocity and the acceleration of the vehicle may oscillate easily, and maintenance of system stability will become impossible. Therefore, we propose the coordination algorithm as shown in Fig. 7.

4. Mobile manipulator control

4.1. Manipulator control

In this subsection, we have developed an assembled robust neuro-fuzzy controller for inexact arm model knowledge given by (1) and (4) to accomplish a certain motion to reach the prescribed goal \( q_{ad} \). To accomplish this task, we first rewrite Eq. (1) as

\[
\tau_a = M_a(q_a)\ddot{q}_a + C_{a1}(q_a, \dot{q}_a, \dot{q}_v)\dot{q}_a + R_a(q_a, q_v)\ddot{q}_v + C_{a2}(q_a, \dot{q}_a, \dot{q}_v)\dot{q}_v + F(\dot{q}_a) + G(q_a).
\]  

Let \( a(t) \in \mathbb{R}^n \) be

\[
a = -\dot{q}_a.
\]  

With regard to differentiating (8) and invoking (7), it is seen that the robot dynamic links are expressed in terms of \( a \) as

\[
M_a(q_a)\dot{a} = -C_{a1}(q_a, \dot{q}_a, \dot{q}_v)a + C_{a2}(q_a, \dot{q}_a, \dot{q}_v)\dot{q}_v + F(\dot{q}_a) + G(q_a) - \tau_a + R_a(q_a, q_v)\ddot{q}_v.
\]  

Adding and subtracting \( M_{a0}(q_a)\dot{a} \) and \( C_{a10}(q_a, \dot{q}_a, \dot{q}_v)a \) yields

\[
M_{a0}(q_a)\dot{a} = -C_{a10}(q_a, \dot{q}_a, \dot{q}_v)a + f_1(x_1) - \tau_a,
\]  

where the nonlinear function \( f_1(x_1) \), which contains the imperfectly known and difficult to determine robot parameters, is defined as

\[
f_1(x_1) = (M_{a0}(q_a) - M_a(q_a))\dot{a} + \begin{pmatrix} C_{a10}(q_a, \dot{q}_a, \dot{q}_v) \\ -C_{a1}(q_a, \dot{q}_a, \dot{q}_v) \end{pmatrix}a + C_{a2}(q_a, \dot{q}_a, \dot{q}_v)\dot{q}_v + F(\dot{q}_a) + G(q_a) + R_a(q_a, q_v)\ddot{q}_v.
\]  

The vector \( x_1 \) required to compute \( f_1(x_1) \) can be defined as \( x_1 = [\dot{q}_a^T \dot{q}_v^T \ddot{q}_a^T \ddot{q}_v^T] \), which can be measured.

Now let \( I_{ad} \) be a value of \( I_a \) that stabilizes the dynamics (10) in the form

\[
M_{a0}(q_a)\dot{a} = -C_{a10}(q_a, \dot{q}_a, \dot{q}_v)a + f_1(x_1) - NK_TI_{ad} + NK_T\eta
\]  

with \( \eta = I_{ad} - I_a \) an error term. Signal \( I_{ad}(t) \) will be selected later. To find the complete error dynamics, differentiate \( Ly \eta \) and substitute from (4) to find
\[ L \ddot{q} = f_2(x_2) + u_d - u_e, \]  
(13)

where the unknown nonlinear motor function is

\[ f_2(x_2) = RI_a + L \dot{I}_{ad} + NK_c \dot{q}_a. \]  
(14)

One may select \( x_2 \) as

\[ x_2 = \begin{bmatrix} I_a^T & \dot{I}_{ad}^T & \dot{q}_a^T \end{bmatrix}. \]

### 4.1.1. Manipulator control structure

A suitable approximation-based controller is given by the desirable value of armature current as

\[ I_{ad} = \frac{1}{K_1} (\hat{f}_1 + K_D a + \tilde{F}_{art} + \gamma_1) \]  
(15)

with \( \hat{f}_1(x_1) \) an estimate of the unknown robot function \( f_1(x_1) \) that is provided by MENN. The signal \( \gamma_1(t) \) is required to compensate for the mismatch between \( \hat{f}_1(x_1) \) and \( f_1(x_1) \). \( K_D \) is a positive definite symmetric matrix.

Using this control in (12), the closed-loop system becomes

\[ M_{a0}(q_a) \ddot{a} = -[K_D + C_{a10}(q_a, \dot{q}_a, \ddot{q}_a)] a + \hat{f}_1(x_1) - \tilde{F}_{art} + \gamma_1, \]  
(16)

where the function approximation error \( \tilde{f}_1 \) is given by

\[ \tilde{f}_1 = f_1 - \hat{f}_1. \]  
(17)

The second step is the design of the voltage control \( u_e \) for the open loop system of (13). The control input \( u_e \) is selected as

\[ u_e = \hat{f}_2 + K_P \eta + \gamma_2 \]  
(18)

with \( K_P > 0 \) a gain matrix and \( \hat{f}_2(x_2) \) an estimate of the unknown function \( f_2(x_2) \) that is provided by robust control. The signal \( \gamma_2(t) \) is required to compensate for the mismatch between \( \hat{f}_2(x_2) \) and \( f_2(x_2) \). Substituting (18) into the open loop dynamics of (13), the closed-loop current perturbation dynamics become

\[ L \ddot{\eta} = \tilde{f}_2(x_2) - K_P \eta, \]  
(19)

where the function approximation error \( \tilde{f}_2 \) is given by

\[ \tilde{f}_2 = f_2 - \hat{f}_2. \]  
(20)

Fig. 8 shows the structure of the proposed control strategy.

### 4.1.2. Recurrent neural network

The MENN, which structure is shown in Fig. 9, is used to approximate nonlinear function \( f_1(x_1) \). In this study we use the sigmoid activation function

\[ \sigma(x_1) = \frac{1}{1 + e^{-x_1}}, \]  
(21)
By collecting the MENN weights $v_{ij}, v_{ip}$ into the matrix of weights $V^T$, and weights $w_i$ into the matrix of weights $W^T$, one can write the MENN equation as

$$y = W^T \sigma(V^T x_1)$$

(22)

with the vector of activation functions defined by $\sigma(z) = [\sigma(z_1) \ldots \sigma(z_n)]^T$ for a vector $z = V^T x_1$ and the vector $x_1 = [1 \ h_1(k-1) \ldots \ h_1(k-j) h_2(k-1) \ldots h_j(k-j) q_1 q_2 \dot{q}_1 \dot{q}_2]$. According to the universal approximation property of neural network, there is a MENN such that

$$f_1(x_1) = W^T \sigma(V^T x_1) + \varepsilon$$

(23)
with the approximation error bounded on compact set by $\|e\| = e_N$ with $e_N$ a known bound. $W$ and $V$ are ideal target weights that give good approximation to $f_i(x_1)$. They are unknown. All we require is the knowledge that they exist; it is not even required for them to be unique. Define the matrix of all ideal MENN weights as

$$Z \equiv \text{diag}\{W, V\}.$$  

(24)

Let a MENN estimate of $f_i(x_1)$ be given by

$$\hat{f}_i(x_1) = \hat{W}^T \sigma(\hat{V}^T)$$

(25)

with $\hat{V}, \hat{W}$ the actual values of the MENN weights given by the tuning algorithm to be specified. Define the hidden-layer output error for a given $x_1$ as

$$\hat{\sigma} = \sigma - \hat{\sigma} = \sigma(V^T x_1) - \sigma(\hat{V}^T x_1).$$

(26)

The Taylor series expansion of $\sigma(x_1)$ for a given $x_1$ may be written as

$$\sigma(V^T x_1) = \sigma(\hat{V}^T x_1) + \sigma'(\hat{V}^T x_1)\hat{V}^T x_1 + O(\hat{V}^T x_1)^2$$

(27)

with Jacobian matrix $\sigma'(\hat{z}) \equiv \frac{d\sigma(z)}{dz}\|z\_t = z$ and $O(z)^2$ denoting term of order two. Denoting $\hat{\sigma}' = \sigma'(\hat{V}^T x_1)$, we have

$$\hat{\sigma} = \sigma'(\hat{V}^T x_1)\hat{V}^T x_1 + O(\hat{V}^T x_1)^2 = \sigma'\hat{V}^T x_1 + O(\hat{V}^T x_1)^2.$$  

(28)

The importance of this equation is that it replaces $\hat{\sigma}$, which is nonlinear in $\hat{V}$, by an expression linear in $\hat{V}$ plus higher-order terms. This will allow us to determine tuning algorithms for $\hat{V}$ in subsequent derivations.

By placing the MENN approximation equation given by (25) into (15), the control input then becomes

$$I_{ad} = \frac{1}{K_1} (\hat{W}^T \sigma(\hat{V}^T x_1) + K_D a + \bar{F}_{art} + \gamma_1)$$

(29)

with $\gamma_1(t)$ a function to be detailed subsequently, providing robustness in the face of higher-order terms in the Taylor series. Using this controller, the closed-loop position error dynamics become

$$M_{a0}(q_a)\dot{a} = -[K_D + C_{a10}(q_a, \dot{q}_a, \ddot{q}_a)]a - \bar{F}_{art} + W^T \sigma(V^T x_1)$$

$$- \hat{W}^T \sigma(\hat{V}^T x_1) + \varepsilon + \gamma_1.$$  

(30)

Adding and subtracting $W^T \sigma$ yields

$$M_{a0}(q_a)\dot{a} = -[K_D + C_0(q_a, \dot{q}_a, \ddot{q}_a)]a - \bar{F}_{art} + \hat{W}^T \sigma - W^T \sigma + \varepsilon + \gamma_1$$

(31)

with $\sigma$ and $\hat{\sigma}$ defining in (26). Adding and subtracting at this point $\hat{W}^T \sigma$ yields

$$M_{a0}(q_a)\dot{a} = -[K_D + C_{a10}(q_a, \dot{q}_a, \ddot{q}_a)]a - \bar{F}_{art} + \hat{W}^T \sigma + \hat{W}^T \sigma + \hat{W}^T \sigma + \varepsilon + \gamma_1.$$  

(32)
The key step is the use at this point of the Taylor series approximation (27) for \( \bar{\sigma} \), according to which the error system is

\[
M_{a0} \dot{a} = -(K_D + C_{10}(q, \dot{q}, \ddot{q}))a - \hat{F}_{\text{art}} + \hat{W}^T (\bar{\sigma} - \sigma' \hat{V}^T x_1) \\
+ \hat{W}^T \sigma' \hat{V}^T x_1 + w + \gamma_1,
\]

where the disturbance terms are

\[
w(t) = \hat{W}^T \sigma' V^T x_1 + W^T O(\hat{V}^T x_1)^2 + \varepsilon.
\]

It is important to note that the MENN reconstruction error \( e(x) \) and the higher-order terms in the Taylor series expansion of \( f_1(x_1) \) all have exactly the same influence as disturbances in the error system; the next bound is required. Its importance is in allowing one disturbance to over bound \( w(t) \) each time by a known computable function.

**Lemma** (Bound on the disturbance term). The disturbance term (34) is bounded according to

\[
\|w\| (\varepsilon_N + \tau_{\text{dist}} + c_5 Z_M) + c_6 Z_M \|\bar{Z}\|_F + c_7 Z_M \|\bar{Z}\|_F \|a\| \text{ or}
\|
w\| C_0 + C_1 \|\bar{Z}\|_F + C_2 \|\bar{Z}\|_F \|a\|
\]

with \( C_i \) known positive constants. Note that \( C_0 \) becomes larger with increases in the MENN estimation error \( \varepsilon \) and the robot dynamics disturbances \( \tau_{\text{dist}} \).

Let the MENN approximation property (23) hold for the function \( f_1(x_1) \) given in (11) with a given accuracy \( \varepsilon_N \) for all \( x_1 \) inside the ball of radius \( r_x > q_M \). Let the initial derivative position error \( a(0) \) satisfy

\[
a(0) < (r_x - q_M)/(c_0 + c_2).
\]

Take the control input of the robot dynamics (7) as (29) with \( K_D \) gain satisfying

\[
K_{D_{\text{min}}} > \frac{(C_0 + \frac{1}{4k} (kZ_M + C_1)^2)(c_0 + c_2)}{r_x - q_M},
\]

where \( C_0 \) and \( C_1 \) are defined in (32), and \( k > 0 \) is a small design parameter. Let the robust term be

\[
\gamma_1 = -K_z(\|\bar{Z}\|_F + Z_M)a,
\]

where \( Z_z > C_2 \). Let MENN weight tuning be provided by

\[
\hat{W} = G_1 \sigma a^T - G_1 \sigma' \hat{V}^T x_1 a^T - kG_1 \|a\| \hat{W},
\]

(39)
\[ \dot{V} = G_2x_1 (\sigma^T \dot{W} a)^T - kG_2 \|a\| \dot{V} \]  

with constant positive diagonal matrices \( G_1 \) and \( G_2 \). Then the derivative position error \( a(t) \) and MENN weight estimates are uniformly ultimately bounded (UUB). The input to (39) and (40) consists of derivative position error \( a(t) \) whereas backpropagation law uses the error between the desired MENN output and the actual MENN output.

4.1.3. Robust term

The control input \( u_c \) is selected as (18) with the signal \( \gamma_2(t) \) defined as

\[ \gamma_2 = \frac{\eta \rho_2^2}{\|\eta\| \rho_2 + \varepsilon_2}, \]  

where \( \varepsilon_2 \) is a positive constant controller gain and small, the bounding positive scalar function \( \rho_2 \) is defined as \( \rho_2 > \|\tilde{f}_2\| = \|f_2 - \tilde{f}_2\| \), where the function approximation error \( \tilde{f}_2 \) is defined in (17).

4.2. Vehicle control

In this subsection, we develop an assembled robust neuro-fuzzy controller for inexact mobile platform model knowledge given by (2) and (5) to accomplish a certain motion. To accomplish this task, we first rewrite Eq. (2) as

\[ E_v \tau_v - A^T \lambda = M_v(q, \dot{q}) \ddot{q} + C_{v1}(q, \dot{q}, \ddot{q}) \dot{q} + C_{v2}(q, \dot{q}, \ddot{q}) \dot{q} + F(\dot{q}) + R_v(q, \dot{q}) \dot{q}. \]  

Differentiating \( A(q) \dot{q} = 0 \), substituting this result in (39), and then multiplying by \( S^T \), we can eliminate the constraint matrix \( A^T(q) \lambda \). The complete equations of motion of the nonholonomic mobile platform are given by

\[ \dot{q}_v = S_v, \]  

\[ S^T M_v \dot{S} \dot{v} + S^T (M_v \dot{S} + C_v S) v + S^T C_{v1} \dot{q}_a + S^T R_v \dot{q}_a + \bar{F}_v(v) = S^T E_v \tau_v, \]  

where \( v(t) \in \mathbb{R}^{n-m} \) is a velocity vector. By appropriate definitions we can rewrite (44) as follow

\[ \bar{\tau}_v = \overline{M}_v(q_a, q_v) \dot{v} + \overline{C}_{v2}(q_a, q_v, \dot{q}_a, \dot{q}_v) v + \overline{C}_{v1}(q_a, q_v, \dot{q}_a, \dot{q}_v) \dot{q}_a \]

\[ + \overline{R}_v(q_a, q_v) \dot{q}_a + \overline{F}_v, \]  

where \( \overline{M}_v(q_a, q_v) = S^T M_v S, \overline{R}_v(q_a, q_v) = S^T R_v(q_a, q_v) \) are symmetric positive definite inertia matrices, \( \overline{C}_{v2}(q_a, q_v, \dot{q}_a, \dot{q}_v) = S^T (M_v \dot{S} + C_v S), \overline{C}_{v1} = S^T C_{v1}(q_a, q_v, \dot{q}_a, \dot{q}_v) \) are the centripetal and coriolis matrices, \( \overline{F}_v(v) \) is the surface friction, and \( \bar{\tau}_v = S^T E_v \tau_v \) is the input torque vector. The fuzzy system provides the torque \( F_v \) that is transformed to the desired velocity \( v_d \).
Now let us define the auxiliary velocity tracking error as

\[ e_v = v_d - v. \]  

Differentiating (46) and using (45), the mobile platform dynamics may be written in terms of the velocity tracking error as

\[ M_v(q_a, q_v) \dot{e}_v + \bar{C}_{v2}(q_a, q_v, \dot{q}_a, \dot{q}_v) e_v + f(x) = \bar{\tau}_v, \]  

where the nonlinear function \( f(x) \) which contains the imperfectly known and difficult to determine robot parameters is defined as

\[ f(x) = M_v(q_a, q_v) v_d + \bar{C}_{v2}(q_a, q_v, \dot{q}_a, \dot{q}_v) v_d + \bar{C}_{v1}(q_a, q_v, \dot{q}_a, \dot{q}_v) \dot{q}_a + \bar{R}_v(q_a, q_v) \dot{q}_a + F_v(v). \]  

The vector \( x \) required to compute \( f(x) \) can be defined as

\[ x = [v^T \ v_d^T \ \dot{v}_d^T \ \dot{q}_a^T \ \dot{\dot{q}}_a^T], \]  

which can be measured.

### 4.2.1. Mobile platform control structure

A suitable control input for velocity following as given by the computed-torque like control

\[ \bar{\tau}_v = \hat{f} + Ke_v - \gamma \]  

with \( K \) is diagonal positive definite gain matrix, and \( \hat{f}(x) \) an estimate of the robot function \( f(x) \) that is provided by the MENN. The robust signal \( \gamma(t) \) is required to compensate the unmodeled unstructured disturbances. Using the control in (49), the closed-loop system becomes

\[ M_v \dot{e}_v + (K + \bar{C}_{v2}) e_v - \bar{f} - \gamma = 0, \]  

where the velocity tracking error is driven by the functional estimation error

\[ \bar{f} = f - \hat{f}. \]  

The structure of our mobile platform controller is shown in Fig. 10.

### 4.2.2. Mobile platform recurrent neural network

In this study, we will use the previous MENN to approximate \( f(x) \) for computing the control in (49). By placing into (49) the MENN approximation equation given by (22), the control input then becomes

\[ \bar{\tau}_v = \hat{W}^T \sigma(\hat{V}^T x) + Ke_v - \gamma \]  

with \( \gamma(t) \) a function to be detailed subsequently that provides robustness in the face of robot kinematics and hither-order terms in the Taylor series. Using this controller, the closed-loop velocity error dynamics becomes

\[ \bar{M} \dot{e}_v = -(K + \bar{C}_{v2}) e_v + W^T \sigma(V^T) - \hat{W}^T \sigma(\hat{V}^T) + \gamma. \]
Adding and subtracting $W^T \tilde{r}$ yields

$$\bar{M} \dot{e}_v = -(K + C_{v_2}) e_v + \tilde{W}^T \tilde{r} + W^T \hat{r} + \epsilon + \gamma$$

with $\hat{r}, \tilde{r}$ defining in (26). Adding and subtracting now $\tilde{W}^T \tilde{r}$ yields

$$\bar{M} \dot{e}_v = -(K + C_{v_2}) e_v + \tilde{W}^T \tilde{r} + \tilde{W}^T \hat{r} + \tilde{W}^T \hat{r} + \epsilon + \gamma.$$  \hfill (55)

The key is the use now of the Taylor series approximation (28) for $\tilde{r}$ according to which the error system is

$$\bar{M} \dot{e}_c = -(K + C_{v_2}) e_v + \tilde{W}^T (\tilde{r} - \hat{r}' \hat{V}^T x) + \tilde{W}^T \hat{r}' \hat{V}^T + w + \epsilon + \gamma,$$  \hfill (56)

where the disturbance terms are

$$w(t) = \tilde{W}^T \hat{r}' \hat{V}^T x + W^T O(\hat{V}^T x) + \epsilon.$$  \hfill (57)

It is important to note that MENN reconstruction error $\varepsilon(x)$ and the higher-order terms in the Taylor series expansion of $f(x)$ all have exactly the same influence as disturbances in the error system. The next bound is required. Its importance is in allowing one to over-bound $w(t)$ at each time by a known computable function.

It remains now to show how to select the tuning algorithms for the MENN weights $\hat{Z}$, and the robust term $\gamma(t)$ so that robust stability and tracking performance are guaranteed. Take the control $\bar{\tau} \in R^r$ for (49) as with robust term
\[ \gamma(t) = -K_2 (\| \widehat{Z} \|_F + Z_M) e_v - e_v \quad (58) \]

and gain \( K_2 > C_2 \) with \( C_2 \) the known constant in (35). Let MENN weight tuning be provided by

\[
\hat{W} = F \hat{\sigma} e_v^T - F \hat{\sigma}' \hat{V}^T x e_v^T - kF \| e_v \| \hat{W},
\]

\[
\hat{V} = Gx(\hat{\sigma}' \hat{W} e_v)^T - kG \| e_v \| \hat{V},
\]

where \( F, G \) are positive definite design parameter matrices, \( k > 0 \) and the hidden-layer gradient or Jacobian \( \hat{\sigma}' \) is easily computed in terms of measurable signals—for the sigmoid activation function it is given by

\[
\hat{\sigma}' \equiv \text{diag} \{ \sigma(\hat{V}^T x) \} [I - \text{diag} \{ \sigma(\hat{V}^T x) t \}].
\]

Then, for large enough control gain \( K \), the velocity tracking error \( e_v(t) \) and the MENN weight estimates \( \hat{V}, \hat{W} \) are UUB. Moreover, the velocity tracking error may be kept as small as desired by increasing the gain \( K \).

5. Simulations

To validate the theoretical concepts and demonstrate the benefits of the proposed fuzzy obstacle avoidance algorithm, a comprehensive simulation study was carried out using a mobile manipulator that consists of a Mitsubishi manipulator, model RV-M1 (Movemaster EX) and a 3-DOF mobile platform whose parameters are listed in Tables 1 and 2, respectively.

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Parameters of the Mitsubishi RV-M1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Degrees of freedom</td>
<td>5 (not including hand)</td>
</tr>
<tr>
<td>Electrical drive system</td>
<td>DC servo motors</td>
</tr>
<tr>
<td>Base</td>
<td>250 mm x 190 mm</td>
</tr>
<tr>
<td>Link operation range and length</td>
<td>Waist ( J_1 ) (revolute joint) 300° (maximum 120°/s) and 300 mm length</td>
</tr>
<tr>
<td></td>
<td>Shoulder ( J_2 ) (revolute joint) 130° (maximum 72°/s) and 215 mm length</td>
</tr>
<tr>
<td></td>
<td>Elbow ( J_3 ) (revolute joint) 110° (maximum 109°/s) and 250 mm length</td>
</tr>
<tr>
<td></td>
<td>Wrist pitch ( J_4 ) (revolute joint) ±90° (maximum 100°/s) and 160 mm length</td>
</tr>
<tr>
<td></td>
<td>Wrist roll ( J_5 ) (revolute joint) ±180° (maximum 163°/s) and 145 mm length</td>
</tr>
<tr>
<td>Maximum payload</td>
<td>1.2 kg including hand</td>
</tr>
<tr>
<td>Maximum path velocity</td>
<td>1000 mm/s</td>
</tr>
<tr>
<td>Weight</td>
<td>19 kg</td>
</tr>
</tbody>
</table>
Simulations were performed on dynamic environment shows in Fig. 11. The environment dimensions are given in meters. The points A, B, C, D, and E are the nodes of graph $G$ obtained by the global road map approach. The distance from the control point F to moving obstacle 11 is the maximum distance of obstacle detection. The vehicle fuzzy local navigator who takes into account the nonholonomic constraints of mobile platform in Reeds and Shepp metric system produces the path between two points the precedent points.

Using intelligent motion planning and robust neuro-fuzzy feedback control proposed in the two previous sections, the simulations shown in Fig. 12 have

<table>
<thead>
<tr>
<th>Parameters of the mobile platform</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimension</td>
<td>$500 , \text{mm} \times 340 , \text{mm}$</td>
</tr>
<tr>
<td>Mass including the driving wheel plus the rotor of its motor</td>
<td>$100 , \text{kg}$</td>
</tr>
<tr>
<td>Electrical drive system</td>
<td>DC servo motors</td>
</tr>
</tbody>
</table>

Fig. 11. Dynamic environment.
been obtained. The mobile platform position, orientation and linear velocity are shown in Figs. 13 and 14, and the arm manipulator joint angles are represented in Fig. 15.
Fig. 15 shows that only the waist and shoulder joint angles vary during the vehicle motion. This intelligent phenomenon is allotted to our proposed algorithm, which determines the variation of joints necessary to the stability of overall electromechanical system during the motion of mobile platform. The mobile manipulator at the start position is shown in Fig. 16. In the entry of narrow passage (see Fig. 17), the manipulator is reconfigured to allow the mobile platform to move stable through a narrow passage. From Fig. 14, it can be deduced that the vehicle reduces its velocity during that operation (from $t = 36$ s to $t = 42$ s). Out of the doorway and before the robot meets a dynamic obstacle shown in Fig. 18 (42 s < t < 58 s), the joint angles of arm manipulator do not vary (refer to Fig. 15). At the same time, Fig. 14 shows that the moving platform linear velocity remains constant. The same phenomenon can be also observed when the robot moves back to the point E of the graph.

It can be observed from Fig. 18 that the robot moves back to the node E of the graph $G$ when it meets a moving obstacle at the control point F shown in Fig. 11. This phenomenon is due to the separation of the bubbles between node E and goal point. With the help of global planner at the point E, the robot chooses an optimal path to reach its goal.

Fig. 19 shows the arm manoeuvre at the goal point. It can be observed that the mobile platform brings the arm at the preferred operating region so that the mobile manipulator remains stable when the manipulator moves its payload.
Fig. 15. Arm joint angles.

Fig. 16. Mobile manipulator at the start position.
As a result of information received by the fuzzy local navigators from different sensors about the form and position of obstacles, the robot successfully completes a complex maneuver avoiding moving and stationary obstacles,
and chooses an optimal path to reach its target. The manipulator is positioned at the optimal configuration and the mobile platform reduces its velocity to help the mobile manipulator to be stable during its motion so that it autonomously avoids obstacles and navigates through a doorway. This phenomenon shows the effective intelligent capability of the proposed motion-planning algorithm to deal with dynamic obstacles and makes our proposed intelligent motion planning very attractive.

6. Conclusion

Using a new intelligent motion control strategy that makes possible the integration of fuzzy obstacle avoidance, multisensor-based motions, and robust recurrent neural network control proposed previously, the simulation results, which need to be validated under real conditions, clearly demonstrate that the proposed strategy is an effective approach. Another important result is the effective intelligent motion capability of the proposed scheme. With the help of its “brain” (fuzzy-decision), which makes decision about any action of the robot, it is shown that the robot autonomously and stably chooses an optimal path to reach its target without collision with unknown and/or moving obstacles in unstructured working areas. This brings a high level of autonomy to the overall system, and makes the use of the controller very attractive for real-time fast and nonstop sensor-based guidance of intelligent mobile manipulators.
Acknowledgment

Dr. MBEDE would like to acknowledge the financial support by the Grant-in-Aid from the Japan Society for the Promotion of Science (JSPS).

References