Neuro-Fuzzy Dynamic Obstacle Avoidance for Autonomous Robot Manipulators*

Jean Bosco MBEDE**, Pierre ELE***
and Huang XINHAN****

This paper presents an integration of fuzzy local planner and modified Elman neural networks (MENN) approximation-based computed-torque controller for motion control of autonomous manipulators in dynamic and partially known environments containing moving obstacles. The navigation is based on fuzzy technique for the idea of artificial potential fields (APF) using analytic harmonic functions. Unlike fuzzy technique, the development of APF is computationally intensive operation. The MENN controller can deal with unmodeled bounded disturbances and/or unstructured unmodeled dynamics of the robot arm. The MENN weights are tuned on-line, with no off-line learning phase required. The stability of the closed-loop system is guaranteed by the Lyapunov theory. The purpose of the controller, which is designed as a Neuro-fuzzy controller, is to generate the commands for the servo-systems of the robot so it may choose its way to its goal autonomously, while reacting in real-time to unexpected events. The proposed scheme has been successfully tested. The controller also demonstrates remarkable performance in adaptation to changes in manipulator dynamics. Sensor-based motion control is an essential feature for dealing with model uncertainties and unexpected obstacles in real-time world systems.

** Key Words: ** Dynamic Obstacle Avoidance, Lyapunov Stability, On-Line Learning, Robot Manipulators, Robust Neuro-Fuzzy Controller

1. Introduction

The obstacle avoidance problem is important for both mobile robots and manipulators. A robust obstacle scheme should be capable of dealing with moving obstacles. For a manipulator, the problem is more complex. Not only must the end effector move to the desired destination without collisions with obstacles, but also the links must avoid collisions. Because this precedent requirement is more restrictive, a strategy that works for manipulators can be easily applied to mobile robots.

In most existing work in autonomous navigation, a solution is attempted by separating the planning and control into two sequential stages. This may have some adverse effects, e.g., algorithm convergence. Robot motion planning with APF considers the problem of motion planning and control simultaneously. The first use of the APF concept for obstacle avoidance was presented by Khatib**. He proposed Force Involving Artificial Repulsion from the Surface (FIRAS, in French), which should be non-negative, continuous and differentiable. However, the potential field introduced, exhibits local minima other than the goal position of the robot. To solve the precedent problem, Volpe and Khosla developed new elliptical potential functions called Superquadric Artificial Potential Functions, which do not generate local

* Received 14th September, 1999
** Humboldt Research Fellow, Institute of Measurement Science, Faculty of Aerospace Engineering, Bundeswehr University Munich, 85577 Neubiberg, Germany. E-mail: mbede@unibw-muenchen.de Laboratoire d'Automatique et de Productique, Département des Génies Industriel et Mécanique, École Nationale Supérieure Polytechnique, Université de Yaoundé I, BP 8390 Yaoundé, Cameroun. E-mail: jmbede@polytech.uninet.cm
*** Laboratoire d'Electronique et de Traitement de Signal, Département des Génies Électriques et Télécommunication, École Nationale Supérieure Polytechnique, Université de Yaoundé I, BP 8390 Yaoundé, Cameroun. E-mail: pele@polytech.uninet.cm
**** Intelligent Control and Robotics Laboratory, Department of Control Science and Engineering, Huazhong University of Science and Technology, 430074 Wuhan, P.R. China. E-mail: xhhuang@hust.edu.cn
minima in physical space. They have shown that superquadric potential fields can be constructed only for simple shapes like square or triangular figures. The problem of local minima remains because the superquadric potential functions do not generate local minima in the workspace but local minima can occur in the Configuration space (C-space), also called the Joint space of the robot. The contributions of Koditschek et al. are worth mentioning because they introduced an analytic potential field in the C-space, without local minima. However the topology of the application ranges is limited to obstacles, which have to be ball- or star-shaped, otherwise no solution can be found. The contributions of Connolly and Kim and Khosla are, in our opinion, the most successful methods concerning robot motion planning with potential fields. They have, simultaneously, used harmonic functions to build a potential field in the C-space without local minima. In the potential field approach to path planning, the development of the APF is computationally intensive operation. It seems very attractive to apply fuzzy logic to reasoning about obstacle avoidance using APF. The proposed fuzzy obstacle avoidance approach entails an attractive force provided by input $e$ (the difference between the goal and the current robot position) and repelling forces provided by input $d$ (the distance between the obstacles and the robot). The sum of all these forces causes a net force to act on the robot. The sign and amplitude of this force are key decision of orientation, speed and acceleration robot motion. Fuzzy decisions, which guide the robot along an obstacle free path in dynamic environments containing moving obstacles, are considered as "brain" of robot system. The proposed algorithm, namely Intelligent Dynamic Motion Planning, is computationally efficient. The processing scheme of fuzzy obstacle avoidance is easily implemented by digital logic circuits. Basic fuzzy logic inference rules are very simple in mathematics. As a result, they can be realised by using simple hardware (microprocessors, operational amplifiers, etc.). The proposed intelligent motion planning is classified as local methods. The motion of robotic manipulators can be broadly categorised into the two classes: global and local methods. Global methods often operate in the C-space, where a single point represents the manipulator configuration. Workspace obstacles are mapped from Cartesian-space into the C-space, and a path is then found in the unoccupied portions of the C-space from the initial point to the goal point that avoids obstacles. Global methods are computationally very expensive, and the computational cost increases rapidly as a function of the number of manipulator joints. On the other hand, local path planning is less computationally demanding than global methods. Local methods can deal with moving obstacles in an unstructured workspace, and can be applied for real time sensor-based motion planning. These attributes make them implementable for on-line collision avoidance at the low execution level of the control hierarchy.

Several autonomous systems have been developed using rule-based methods to control the motion of robot manipulators. Tsoukalas et al. presented a neuro-fuzzy methodology for a robot to navigate in dynamic environments. Ding and Li solved the problem of obstacle avoidance for a redundant manipulator by using a fuzzy logic system. All these systems do not consider the structured and unstructured uncertainties, and/or are limited to the avoidance of stationary obstacles.

Unlike motion planning, a robust fuzzy obstacle avoidance scheme should also be capable of dealing with structured and unstructured uncertainties. To accomplish this task, we proposed an integration of robust controller and MENN controller to approximate imperfectly known and/or unmodeled robot dynamic parameters. 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 modified Elman neural network is a recurrent neural network used by Pham and Liu 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 arm movements based on the trajectory planning given by the fuzzy controller. To keep the weights of the MENN from growing without bounds, we use a robust term in the control law. The closed-loop stability of the system is rigorously tested. In real-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: Section 2 reviews some basic concepts of motion planning through artificial potential functions. After some definitions and requirements, harmonic functions are introduced. We also present a dynamic model of manipulator. Section 3 proposes a fuzzy obstacle avoidance, robust and MENN controller. The stability of the closed-loop system is proved by Lyapunov's theory. Section 4 presents some simulation results. Finally, Section 5 gives some concluding
2. Preliminaries

2.1 The robot model and its properties

The dynamics of a robot manipulator are well understood (see e.g.,) and are given by

$$\tau = M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + G(q) + F(q) + \tau_d$$  \hspace{1cm} (1)

where $q, \dot{q}, \ddot{q} \in \mathbb{R}^n$ denotes the link position, velocity, and acceleration vectors, respectively, $M(q) \in \mathbb{R}^{n \times n}$ the inertia matrix, $C(q, \dot{q}) \in \mathbb{R}^{n \times n}$ the coriolis/centripetal matrix, and $G(q) \in \mathbb{R}^n$ the gravitational torque vectors, $F(q) \in \mathbb{R}^n$ represents the friction terms, $\tau \in \mathbb{R}^n$ is joint drive torques (and/or forces), and $\tau_d \in \mathbb{R}^n$ are the vectors of any generalised input due to disturbances or unmodeled dynamics.

There are uncertainties in $M(q)$ and $C(q, \dot{q})$ due to the unknown load on the manipulator. We assume the following bounds on the uncertainties:

**Assumption 2.1.1 (Bounds on inertia matrix):** There exist positive definite matrices $M_0(q)$ and $M_{\text{max}}(q)$ such that

$$0 < M_{\text{min}}(q) \leq M(q) \leq M_{\text{max}}(q).$$  \hspace{1cm} (2)

**Assumption 2.1.2:** There exist $C_0(q, \dot{q})$ and a nonnegative function $C_{\text{max}}(q, \dot{q})$ such that

$$\|C(q, \dot{q}) - C_0(q, \dot{q})\| \leq C_{\text{max}}(q, \dot{q}).$$  \hspace{1cm} (3)

**Assumption 2.1.3 (Bounded disturbances):** The disturbances are bounded so that $|\tau_d| = r_{\tau_{\text{max}}}.$

The following property can be exploited to facilitate control system design.

**Property 2.1.1 (Skew–symmetry):** The inertia and centripetal–coriolis matrices have the following property:

$$q^T(M_0(q) - 2C_0(q, \dot{q})) \dot{q} = 0, \hspace{1cm} q \in \mathbb{R}^n,$$  \hspace{1cm} (4)

where $M_0(q) - 2C_0(q, \dot{q})$ is a skew symmetric matrix.

2.2 Motion planning through APF

2.2.1 General concepts

The APF approach may be described as follows. If $q_d$ designates the goal position, the guidance of the robot with respect to n obstacles $O_i$ ($i = 1, \ldots, n$), can be achieved by subjecting it to the APF:

$$\varphi_{\text{err}}(q) = \varphi_{\text{a}}(q) + \sum_{i=1}^{n} \varphi_{\text{o}}(q),$$  \hspace{1cm} (5)

where $\varphi_{\text{err}}(q)$ is the total strength of the APF at the point $q$, $\varphi_{\text{a}}(q)$ the APF strength contribution from the attractive goal and $\varphi_{\text{o}}(q)$ is the contribution from the $i$th repulsive obstacle. This field causes the following artificial force to act on the robot:

$$F_{\text{err}}(q) = F_{\text{a}}(q) + \sum_{i=1}^{n} F_{\text{o}}(q),$$  \hspace{1cm} (6)

where $F_{\text{a}}(q) = -\nabla \varphi_{\text{a}}(q)$ and $F_{\text{o}}(q) = -\nabla \varphi_{\text{o}}(q)$.

For illustration, consider the two-dimensional obstacle avoidance situation illustrated in Fig. 1, in which the robot is constrained to move. Consider the link of robot to be a negatively charged particle trying to reach the positively charged goal position. If there were no obstacles in the environment, it would be a simple matter of moving in a straight line between the start and the goal. However, when there are obstacles present, their negative charge repels the links of robot.

The major problem in the potential field motion planning approach is the occurrence of local minima in the potential field, which cause the untimely termination of the motion of the robot. A robust APF should have no local minima. These kinds of robust artificial potential fields include the employment of the so-called harmonic functions. The harmonic functions attain their extreme values at the boundary of the domain.

2.2.2 Analytic harmonic functions

Kim and Khoosal introduced an artificial potential approach based on harmonic functions to guide a robot in the C-space. The harmonic functions are functions, which satisfy the following equation

$$\nabla \varphi(q) = 0,$$  \hspace{1cm} (7)

which is called the Laplace equation.

The most important properties of harmonic functions are that they are free of local minima and that any linear combination of two harmonic functions is also harmonic function. The Laplace equation (7) can be written in general polar co-ordinates as

$$\nabla^2 \varphi = \frac{\partial^2 \varphi}{\partial r^2} + \frac{n-1}{r} \frac{\partial \varphi}{\partial r} + \text{angular terms}.$$  \hspace{1cm} (8)

We assume that the harmonic function $\varphi$ is a function of $r$ only, the angular terms are zero. After rearranging and integrating with respect to $r$ Eq. (8) becomes

$$\frac{\partial \varphi}{\partial r} = -\frac{c}{r^{n-1}}.$$  \hspace{1cm} (9)

For $n=2$, the solution of Eq. (9) is

$$\varphi = c \log r + C_i.$$  \hspace{1cm} (10)

If $n>2$, the solution is

![Fig. 1 Two-dimensional APF](image-url)
where $r = \sqrt{(q_x - q_{oa})^2 + (q_y - q_{oa})^2 + \cdots + (q_n - q_{oa})^2}$, with $q_o$ being an arbitrary point.

Depending on the sign of the constant scalar $c$, Eqs.(10) and (11) can be used to represent a source or a sink. A source can be used to represent a point obstacle at $q_o$ and a sink can be used to represent the goal position at $q_o = q_a$.

The obstacles in C-space with an arbitrary shape can be represented by a number of panels (for more details see\textsuperscript{8}).

2.2.3 Sensor-based obstacle avoidance

Generally, a goal is not directly accessible or even identifiable and navigation must be an incremental process. The potential must depend directly on perceptual data and the generation of the displacement must be done incrementally as a function of the evolution of the environment. In order to avoid obstacles effectively, a system must be informed about the possible positions of obstacles and their likely future positions. This is the case for the repulsive potential associated with the distance to the obstacle (refer Fig. 2). In real-time, and because of the maximum range of the ultrasonic sensors\textsuperscript{10}, the distance to the point of contact is not always determined. Nevertheless, in such a case, we have the information that the contact is at a distance greater than the maximum distance related to the sensor. This maximum distance generates an attraction force producing an elementary displacement. As long as the object is not detected, the norm of this force remains constant and consequently, it is not a conservative force. The effect is the same if the object representing the contact is itself moving.

3. Robust Neuro-Fuzzy Motion Control

3.1 Intelligent dynamic motion planning

In this subsection, we realize a collision avoidance based on Fuzzy decision, which is considered as the “brain” of robot.

Before describing a fuzzy decision, we first develop a design procedure consisting of the selection of membership functions and the establishment of a rule base for fuzzy system.

3.1.1 Fuzzification

The fuzzy system has as inputs: the distance $d$ between the link and the nearest obstacle, and the position error $e$. The output of this controller is the force $F_{an}$, which is required to be bounded: $|F_{an}| < \infty$. All these variables can be positive as well as negative, thus, they do not only inform about the magnitude, but also about the sign of displacement.

The position error $e$ 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. 3 where $\xi$ denotes the maximum bound of the $w$-space. The distance $d$ 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 link, the distance $d$ is negative, and when it is in right or front of the arm, the distance $d$ is
positive. Its fuzzy membership functions are asymmetric and shown in Fig. 4 where \( \delta \) is the maximum ultrasonic range.

3.1.2 Rule base  The rule base is generalized as follows:

\[ R^i : \text{if } e(k) < \mu_e(e(k)) \text{ and } \ldots \text{ and } e(k-n+1) < \mu_e(e(k-n+1)) \text{ and } d(k) < \mu_d(d(k)) \text{ and } \ldots \text{ and } d(k-m+1) < \mu_d(d(k-m+1)) \text{ then } F(k+1) = r^i, \]

where \( R^i (i = 1, 2, \ldots, l) \) denotes the \( i \)-th implication, \( l \) is the number of fuzzy rules, \( r^i \) is the output from the \( i \)-th implication, \( n \) is the number of input variable \( e \), and \( m \) is the number of input variable \( d \). Our thirty rule bases are arranged into a look-up table and shown in Table 1.

The inputs \( \mu(e) \) and \( \mu(d) \) represent the fuzzy sets, which indicate the distance between robot and obstacle, and the position error, respectively. The outputs of the base are \( F \) which describe the torque output and they are partitioned into nine fuzzy sets: left very big (LVB), left big (LB), left small (LS), left very small (LVS), zero (Z), right very small (RVS), right small (RS), right big (RB), and right very big (RBB).

3.1.3 Defuzzification  The following defuzzification formula is used.

\[ \vec{F}_{arr} = \frac{\sum F_i w^i}{\sum w^i}, \]

where the weight is

\[ w^i = \prod_{k=1}^{l} \mu_e(e(k-p+1)) \times \prod_{h=1}^{n} \mu_d(d(k-h+1)). \]

\( \vec{F}_{arr} \in \mathbb{R}^2 \) is the approximated APF output \( \vec{F}_{arr} \).

3.1.4 Decision  The sign of the torque \( \vec{F}_{arr} \) gives the orientation of the link displacement, e.g. for a planar robot, when \( \vec{F}_{arr} \) is positive, the link moves to the left; when it is negative, the link moves to the right. The magnitude of this torque provokes the acceleration and deceleration motion of the link.

3.2 Control Design  The control objective is to develop a robust and intelligent controller for robot dynamics given by Eq. (1) to accomplish a certain motion to reach the prescribed goal \( q_a \). The trajectory of the robot is not known or calculated in advance. To accomplish this task, we first define the position error \( e(t) \in \mathbb{R}^n \) and its first derivative \( \dot{e}(t) \in \mathbb{R}^n \) by:

\[ e = q_a - q, \]
\[ \dot{e} = \dot{q} = a. \]

With regard to differentiating Eq. (16) and invoking Eq. (1), it is seen that the robot dynamics are expressed in terms of \( a \) as

\[ M(q) \ddot{a} = -C(q, \dot{q}) a + G(q) + F(\dot{q}) + \tau_a - \tau. \]

Adding and subtracting \( M(q) \ddot{a} \) and \( C(q, \dot{q}) a \) yields

\[ M(q) \ddot{a} = -C(q, \dot{q}) a + f(x) + \tau_a - \tau, \]

where the nonlinear robot function \( f(x) \) is defined as

\[ f(x) = (M(q) - M(q)) \ddot{a} + (C(q, \dot{q}) - C(q, \dot{q})) a + G(q) + F(\dot{q}). \]

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

\[ x = [a \tau_1 \tau_2 \tau_3] \tau, \]

which can be measured.

Function \( f(x) \) contains the imperfectly known and difficult to determine robot parameters.

3.2.1 Controller structure  A suitable approximation-based controller is given by the computed-torque type control

\[ \tau = \ddot{f} + \vec{F}_{arr} + K_{oo} a - \gamma, \]

with \( f(x) \) an estimate of the robot function \( f(x) \) that is provided by MENN. \( \vec{F}_{arr} \) is an artificial force, provided by the fuzzy controller, and acting on the robot. The signal \( \gamma(t) \) is required to compensate for the unmodeled unstructured disturbances. \( K_{oo} \) is a positive definite symmetric matrix.

Using this control in Eq. (18), the closed-loop system becomes

\[ M(q) \ddot{a} = -[K_{oo} + C(q, \dot{q})] a - \vec{F}_{arr} + \dot{f} + \tau_a + \gamma, \]

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

\[ \dot{f} = f - \vec{F}_{arr}. \]

Figure 5 shows the structure of this control strategy. It is important to note that there is one controller at each joint.

3.2.2 MENN approximation system  According to the universal approximation property of Neural Networks, we use a MENN shown in Fig. 6 in our design for the reasons given in section 1. The MENN output \( y \) is determined by the formula

\[ y(k) = \sum_{i=1}^{n} w_i \left[ \frac{\sum_{j=1}^{n} a_{i,j} - v_{i,j} x_{j}(k-j)}{\sum_{j=1}^{n} v_{i,j} x_{j}(k-j) + \theta_{i}(k)} \right] + \theta_{i}(k). \]

---

Table 1  Rule base

<table>
<thead>
<tr>
<th>Rule base</th>
<th>F</th>
<th>F</th>
<th>F</th>
<th>F</th>
<th>F</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>BN</td>
<td>LS</td>
<td>RB</td>
<td>RBB</td>
<td>LVB</td>
<td>LB</td>
<td>LB</td>
</tr>
<tr>
<td>SN</td>
<td>LS</td>
<td>RVS</td>
<td>RB</td>
<td>LB</td>
<td>LB</td>
<td>LS</td>
</tr>
<tr>
<td>Z</td>
<td>Z</td>
<td>Z</td>
<td>RS</td>
<td>LS</td>
<td>Z</td>
<td>Z</td>
</tr>
<tr>
<td>SP</td>
<td>RS</td>
<td>RB</td>
<td>RB</td>
<td>LVB</td>
<td>RS</td>
<td></td>
</tr>
<tr>
<td>BP</td>
<td>RB</td>
<td>RB</td>
<td>RBB</td>
<td>LVB</td>
<td>LB</td>
<td>RS</td>
</tr>
</tbody>
</table>
function given in Eq. (19), is:
\[ f(x) = y + e = W^T \sigma(V^T x) + e, \]
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 \( f(x) \). 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
\[ Z = \text{diag}(W, V). \]
Let a MENN estimate of \( f(x) \) be given by
\[ \hat{f}(x) = \hat{W}^T \sigma(\hat{V}^T x), \]
with \( \hat{V}, \hat{W} \) the actual values of the MENN weights given by the tuning algorithm to be specified. Note that \( \hat{V}, \hat{W} \) are estimates of the ideal weights values and define the weights estimation errors as
\[ \tilde{V} = V - \hat{V}, \quad \tilde{W} = W - \hat{W}, \quad \tilde{Z} = Z - \hat{Z}. \]
Define the hidden-layer output error for a given \( x \) as
\[ \tilde{\sigma} = \sigma - \sigma(\hat{V}^T x) = \sigma(\hat{V}^T x) - y. \]
The Taylor series expansion of \( \sigma(x) \) for a given \( x \) may be written as
\[ \sigma(\hat{V}^T x) = \sigma(\hat{V}^T x) + \sigma'(\hat{V}^T x) \tilde{V}^T x + O(\tilde{V}^T x)^3, \]
with Jacobian matrix \( \sigma'(z) = \frac{d\sigma(z)}{dz} \bigg|_{z=x} \) and \( O(z)^3 \) denoting term of order two. Denoting \( \tilde{\sigma}' = \sigma'(\hat{V}^T x) \), we have
\[ \tilde{\sigma}' = \sigma'(\hat{V}^T x) \tilde{V}^T x + O(\tilde{V}^T x)^3. \]
The importance of this equation is that it replaces \( \tilde{\sigma} \), which is nonlinear in \( \tilde{V} \), by an expression linear in \( \tilde{V} \) plus higher-order terms. This will allow us to determine tuning algorithms for \( \tilde{V} \) in subsequent derivations.

Different bounds may be put on the Taylor series higher-order terms depending on the choice of the activation functions \( \sigma(\cdot) \).

The following assumptions always hold in practical applications.

**Assumption 3.3.1** (Bounded ideal MENN weights) : On any compact subset of \( \mathbb{R}^n \), the ideal MENN weights are bounded by known positive values so that \( |V|^2 \leq V_u, |W|^2 \leq W_u \), or \( |Z|^2 \leq Z_M \) with \( V_u, W_u, Z_M \) known, and \( \|\cdot\|_F \) the Frobenius norm.

**Definition 3.3.1** (Frobenius norm) : Given a matrix \( A = [a_{ij}] \), the Frobenius norm is defined as the root of the sum of squares of all elements:
\[ \|A\|^2_F = \text{tr}(A^T A) = \sum_{i,j} a_{ij}^2, \]
with \( \text{tr}(\cdot) \) the matrix trace (i.e. sum of diagonal elements). Though the Frobenius norm is not an induced norm, it is compatible with the vector 2-norm.
so that
\[ \|Ax\| \leq \|A\| \|x\|. \]  \hspace{1cm} (35)

**Lemma 3.3.1 (Bound on MENN Input x):** For each time \( t, z(t) \), in Eq.(17) is bounded by
\[ |x| \leq q_0 + c_0(\|a\| + c_1(\|a\| + c_2(\|a\|) \leq \alpha + c_0 \|a(\|a)\| \]  \hspace{1cm} (36)
for computable positive constants \( c_0 \).

**Lemma 3.3.2 (Bounds on Taylor series higher-order terms):** For sigmoid functions, the higher-order terms in the Taylor series are bounded by
\[ \|O(\tilde{V}\tilde{z})\| \leq \alpha + c_0 \|\tilde{V}\| + c_0 \|a(\|a)\| \]  \hspace{1cm} (37)
for computable positive constants \( c_0 \).

3.2.3 Robust term system
We will use a MENN to approximate \( f(x) \) for computing the closed-loop system in Eq.(22).

By placing the robot nonlinear function given by Eq.(27) and the MENN approximation equation given by Eq.(30) into Eq.(22), the closed-loop position error dynamics become
\[ M_0(q) \tilde{a} = -[K_0 + C_0(a, q)] \tilde{a} - \tilde{F}_{a} + W^T \sigma(\tilde{V}^T \tilde{x}) - W^T \sigma(\tilde{V}^T \tilde{x}) + (\varepsilon + c_0) + \gamma \]  \hspace{1cm} (38)
with robust term \( \gamma(t) \) a function to be detailed subsequently, providing robustness in the face of higher-order terms in the Taylor series.

Adding and subtracting \( W^T \sigma \) in Eq.(38) yields
\[ M_0(q) \tilde{a} = -[K_0 + C_0(a, q)] \tilde{a} - \tilde{F}_{a} + W^T \tilde{a} - W^T \sigma(\tilde{V}^T \tilde{x}) + (\varepsilon + c_0) + \gamma \]  \hspace{1cm} (39)
with \( \varepsilon \) and \( \sigma \) defining in Eq.(31). Adding and subtracting at this point \( \tilde{W} \) yields
\[ M_0(q) \tilde{a} = -[K_0 + C_0(a, q)] \tilde{a} - \tilde{F}_{a} + W^T \tilde{a} + W^T \sigma(\tilde{V}^T \tilde{x}) + (\varepsilon + c_0) + \gamma \]  \hspace{1cm} (40)
The key step is the use at this point of the Taylor series approximation (32) for \( \tilde{a} \), according to which the error system is
\[ M_0 \tilde{a} = -[K_0 + C_0(a, q)] \tilde{a} - \tilde{F}_{a} + W^T \tilde{a} + W^T \sigma(\tilde{V}^T \tilde{x}) + \tilde{w}(t) \]  \hspace{1cm} (41)
where the disturbance terms are
\[ \tilde{w}(t) = W^T \tilde{a} + W^T \varepsilon(\tilde{V}^T \tilde{x}) + W^T \varepsilon(\tilde{V}^T \tilde{x}) + (\varepsilon + c_0) \]  \hspace{1cm} (42)
It is important to note that the MENN reconstruction error \( \varepsilon(x) \), the disturbance \( \varepsilon(t) \), 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 disturbance to overbound \( \tilde{w}(t) \) each time by a known computable function.

**Lemma 3.3.3 (Bound on the disturbance term):** the disturbance term (42) is bounded according to
\[ \|\tilde{w}\| \leq C_0(\tilde{Z}\|r\|_1 + C_0(\tilde{Z}\|r\|_1 \]  \hspace{1cm} (43)
with \( C_0 \) known positive constants. Note that \( C_0 \) becomes larger with increases in the MENN estimation error \( \varepsilon \) and the robot dynamics disturbances \( \varepsilon(t) \).

Let the MENN approximation property (27) hold for the function \( f(x) \) given in Eq.(19) with a given accuracy \( \varepsilon \), for all \( x \) inside the ball of radius \( r_x > q_x \).

Let the initial derivative position error \( a(0) \) satisfy
\[ a(0) \leq \gamma(r_x - q_x)/(c_0 + c_0) \]  \hspace{1cm} (44)
Let the ideal target MENN weights be bounded as in Assumption 3.3.1. Take the control input of the robot dynamics (1) as (21) with \( K_0 \) gain satisfying
\[ K_{D_{\min}} > \frac{C_0 + c_0}{4k} h(z_n + C_n)^2 (c_0 + c_0) \]  \hspace{1cm} (45)
where \( C_n \) and \( C_n \) are defined in Eq.(43), and \( k > 0 \) is a small design parameter. Let the robust term be
\[ \gamma = -K_0(\|z\| + q_m) \]  \hspace{1cm} (46)
where \( K_0 > C_n \). Let MENN weight tuning be provided by
\[ \tilde{W} = G_1(\tilde{a}^T - G_1(\tilde{a}^T + hG_1\|a\|\tilde{W}), \]  \hspace{1cm} (47)
\[ \tilde{V} = G_2(\tilde{a}^T - hG_2\|a\|\tilde{V}, \]  \hspace{1cm} (48)
with constant positive diagonal matrices \( G_1 \) and \( G_2 \). Derive the derivative position error \( a(t) \) and MENN weight estimates are uniformly ultimately bounded (UUB).

The input to Eqs.(47) and (48) consists of derivative position error \( a(t) \) whereas backpropagation law uses the error between the desired MENN output and the actual MENN output.
to a standard Lyapunov theory and LaSalle extension, this demonstrates the UUB stability of both $\|a\|$ and $\|Z\|$. Note that $\|a\|$ can be kept arbitrarily small by increasing the gain $K_{x_{min}}$ in Eq. (54). Finally, the right-hand sides of Eqs. (54), (55) can be taken as practical bounds on $a(t)$ and the MENN weight estimation errors respectively.

4. Simulations

A comprehensive simulation study was carried out using two-DOF experimental robot manipulator systems (Fig. 7).

The payload with maximum mass $m_p \max = m_{pd} = 2$ kg was attached to the end-effector. The mass of the actuator of the lower arm is $m_a = 4$ kg. The inertia matrix was

$$M(q) = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix},$$

(56)

where

$$M_{11} = f_1 + m_1 r_1^2 + f_2 + m_2 (h_2^2 + r_2^2) + 2 l_1 h_2 \cos(q_2),$$

(57)

$$M_{12} = M_{21} = f_1 + m_1 r_1^2 + f_2 + m_2 r_2^2 + l_1 h_2 \cos(q_2),$$

(58)

$$M_{22} = f_2 + m_2 r_2^2 + m_1 l_2^2.$$  

(59)

In Eq. (57) the moment of inertia of the housing of the elbow joint has been disregarded because this term is small compared to the term $m_a l_1^2$. $f_1$ and $f_2$ are the moment of inertia of upper and lower arm, respectively, with $f_1 = m_1 r_1^2/3$ and $f_2 = m_2 r_2^2/3$. The load $m_{po} = 2$ kg is unknown. The Coriolis/centripetal matrix is

$$C(q, \dot{q}) = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix},$$

where $\beta = (m_3 h_3 r_3 + m_3 h_5 l_2) \sin(q_3)$, $C_{11} = -\dot{q}_2$, $C_{12} = -(\dot{q}_1 + \dot{q}_2)$, $C_{21} = -\dot{q}_1$, and $C_{22} = 0$. The value of the parameters are: $m_1 = 1.09$ kg, $m_2 = 1.12$ kg, $r_1 = 0.132$ m, $r_2 = 0.6448$ m, $l_1 = 1.0$ m, $l_2 = 1.3$ m.

With the previous values, we can compute $M(q)$ and $C(q, \dot{q})$ as follows:

$$M(q) = \begin{bmatrix} 11.146 + 6.644 \cos(q_2) & 4.001 + 3.322 \cos(q_2) \\ 4.001 + 3.322 \cos(q_2) & 4.001 \end{bmatrix},$$

and

$$C(q, \dot{q}) = 3.322 \sin(q_2) \begin{bmatrix} -\dot{q}_2 \\ -\dot{q}_1 - \dot{q}_2 \end{bmatrix}.$$  

The subscript 1 refers to the upper arm, and the lower arm is index 2. The controller gain is selected as

$$K_c = \begin{bmatrix} 50 \\ 0 \\ 0 \end{bmatrix}.$$  

The value of fuzzy rule base is shown in Table 2.

For the MENN, we selected the sigmoid activation functions with one output, $n = 8$ hidden-layer neurons, $m = 8$ inputs, $a = 0.6$, $k = 0.1$ and $G_1 = G_2 = \text{diag}(10, 10)$.

Two cases were used to demonstrate and evaluate the effectiveness of the proposed fuzzy control strategy for robot manipulator navigation. This was done using APF in dynamic and imperfectly known environments, and with static and moving obstacles.

Case 1: Static Obstacles

By referring to the robot parameters described above, using the fuzzy control method with the rule base shown in Table 2 and the fuzzy membership functions shown in Figs. 3 and 4 where $\xi = l_1 + l_2 = 2.3$ m, $\delta = 2.8$ m, $\gamma = 0.4$ m, and $d_a = 20$ mm, the following simulations (Fig. 8) was obtained. The mass of payload was assumed to be $m_p = 0.5$ kg. The initial and desired joint angles are $q_1(0) = 91.6732$, $q_2(0) = 5.7296$, $q_{1d} = -73.5211$, $q_{2d} = 40.1070$ (in deg.).

It can be deduced that owing to the information received by the fuzzy controller from the sensors about the obstacle position, the robot autonomously avoids the obstacles and chooses its way to reach the aimed target. This is the case of repulsive potential.

The simulation results (Fig. 9) show the joint angle, the joint velocity, the joint torque, the position error and the linear acceleration of the links, and the distance between link 2 and the nearest obstacle.

It can be observed that while avoiding the obstacles, the linear acceleration of the links is constant and equal to the gravitational acceleration of the earth, and the joint velocity decreases. After avoiding

![Fig. 7 Model of Two-DOF experimental robot system](image)

---

Table 2: Value of rule base

<table>
<thead>
<tr>
<th>$\mu$</th>
<th>TL/TH</th>
<th>ML/MO</th>
<th>CL/CH</th>
<th>CL/CP</th>
<th>ER/EP</th>
<th>FR/FP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>-0.3</td>
<td>0.5</td>
<td>0.0</td>
<td>0.3</td>
<td>0.3</td>
<td></td>
</tr>
<tr>
<td>0.9</td>
<td>-0.3</td>
<td>0.5</td>
<td>0.0</td>
<td>0.3</td>
<td>0.3</td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td>-0.3</td>
<td>0.5</td>
<td>0.0</td>
<td>0.3</td>
<td>0.3</td>
<td></td>
</tr>
<tr>
<td>0.7</td>
<td>-0.3</td>
<td>0.5</td>
<td>0.0</td>
<td>0.3</td>
<td>0.3</td>
<td></td>
</tr>
<tr>
<td>0.6</td>
<td>-0.3</td>
<td>0.5</td>
<td>0.0</td>
<td>0.3</td>
<td>0.3</td>
<td></td>
</tr>
</tbody>
</table>

---

*JSME International Journal*
the collision, the linear acceleration decreases and then increases, but the joint velocity increases. The graph of the position error shows that at $q = q_d$, our system is stable. Similarly for the limitation range of the ultrasound sensors, the maximum distance to the contact was 2.8 m. This is imposed by the Fuzzy system (refer Fig. 4), and also represents the distance greater than the maximum distance between link and nearest obstacle.

The simulation results of Neuro-fuzzy controller (Fig. 10) show the Fuzzy and MENN output, and the performance of the MENN.

![Diagram](image)

**Fig. 8** Robot Navigation with static obstacles

It can be observed that the fuzzy controller is a planner: when its output is negative, the link moves to the right, and the positive value of this output produces a left displacement of the link. Any variation of this output also produces the linear acceleration or deceleration of the link. Data describing the mechanics of the manipulator become available only after movements have been processed. During this period of data acquisition and training period, the trajectory planning performance will gradually improve. This is one of the major differences of the proposed MENN controller to other MENN based techniques, which utilize off-line training methods.

**Case 2**: Dynamic Obstacles.

By adding dynamic and unknown obstacles, and taking the mass of payload $m_p = 1$ kg, while keeping the other simulation parameters for case 1, the following simulation (Fig. 11) has been obtained. The initial and desired joint angles are $q_1(0) = 91.6732$, $q_2(0) = 5.7296$, $q_3 = -73.5211$, $q_4 = 11.4592$ (in deg.).

It can be observed that owing to the position of the obstacles, the manoeuvre is not easy for the robot, but it successfully avoids the moving and static obstacles. The velocity of the moving obstacle is constant. Thus the manipulator automatically reduced its velocity to avoid collision.

The simulation results (Fig. 12) show the joint angle, the joint velocity, the joint torque, the position error and the linear acceleration of the links, and the

![Diagram](image)

**Fig. 9** Simulation results of robot motion with static obstacle
Fig. 10 Simulation results of Neuro-fuzzy controller

Fig. 11 Robot Navigation with moving and unknown obstacles

distance between link 2 and the nearest obstacle.

Owing to the difficult maneuver of the robot, the linear acceleration of the links is maintained approximately constant and equal to the gravitational acceleration of the earth. When the links avoid the obstacles, their joint velocity is reduced, and augments after avoiding the collision. By comparing Case 1, the time taken by the robot manipulator to reach a goal increases.

The simulation results of Neuro-fuzzy controller (Fig. 13) show the Fuzzy and MENN output, and the performance of the MENN.

It can be observed that the change in the payload mass in the manipulator dynamics can be accounted by the MENN controller. When the system is operating, perturbations in system dynamics are compensated effectively by the robust structure of the controller.

5. Conclusion

A new, stable, robust and intelligent controller that makes possible the integration of the fuzzy controller and modified Elman neural networks (MENN), an approximation-based computed-torque controller has been proposed for autonomous navigation of robot manipulators in dynamic and partially known environments, and with moving obstacles. Using this new Neuro-fuzzy feedback servo-control scheme, the simulation results clearly demonstrated that the proposed strategy is an effective approach to the control of robotic manipulators. Another important result is the effective autonomous motion capability of the proposed scheme. It is shown that the robot autonomously avoids the obstacles and chooses its way to its goal. The other advantage of the proposed controller is that it does not require a perfect knowledge of the robot manipulator parameters, e.g., friction is very difficult to model by conventional techniques, because the MENN learns them on-line. This brings a high level of autonomy to the overall system, and makes the use of the controller very attractive for
Fig. 12 Simulation results of arm navigation among dynamic and unknown obstacles

Fig. 13 Simulation results of neuro-fuzzy controller

real time applications where manipulator dynamics can experience parameter variations, load changes and any possible external disturbances.

Acknowledgements

The authors would like to acknowledge the cooperation of Dr. Kaspar Althöfer of Department of
Mechanical Engineering of King's College London, UK. Dr. J.-B. Mbede is also grateful for the financial support provided by Research Fellowships from Alexander von Humboldt Foundation.

References


(7) Connolly, C.J., Harmonic Functions as a Basis for Motor Control and Planning, Ph.D. Dissertation, Department of Computer Science, University of Massachusetts Amherst, Massachusetts, USA, (1994).


