Neurocomputing 137 (2014) 34–46

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Artificial neural networks aided solution to the problem of geometrically bounded singularities and joint limits prevention of a three dimensional planar redundant manipulator

Samer Yahya [a,*], M. Moghavvemi [b,c], Haider A.F. Mohamed [d]

[a] Department of Mechanical, Materials and Manufacturing Engineering, The University of Nottingham Malaysia Campus, Malaysia
[b] Center of Research in Applied Electronics (CRAE), University of Malaya, Malaysia
[c] Faculty of Electrical and Computer Engineering, University of Tehran, Iran
[d] Department of Electrical and Electronic Engineering, The University of Nottingham Malaysia Campus, Malaysia

A R T I C L E I N F O

Article history:
Received 17 January 2013
Received in revised form
24 October 2013
Accepted 3 November 2013
Available online 20 February 2014

Keywords:
Redundant manipulator
Kinematics
Dynamics
Control
Neural networks

A B S T R A C T

This paper presents a neural network based on a nonlinear dynamical control of a three-dimensional six degrees of freedom planar redundant manipulator. An artificial controller is used for the computation of fast inverse kinematics, and is effective on geometrically bounded singularities and joint limits prevention of redundant manipulators. A comparison between the results of a multilayer back propagation and the radial basis function neural network has been carried out, and the results show that the radial basis function of neural networks is more attractive due to their fast training, simplicity, and convergence rate. The radial basis function neural network has been used to estimate the centrifugal and gravitational effects of the joints, while the end-effector follows a desired path.

1. Introduction

A robotic manipulator that have more degrees of freedom than the number of independent task coordinates is known as a redundant manipulator. Due to these extra degrees of freedom, the robot is capable of performing auxiliary tasks such as singularity and joint limits avoidance. Generally, redundancy specifies the internal movement of the robot, without affecting the trajectory of the end-effector,while permitting the robot to react in a better way to itssurroundings. The important role of redundant robots in space and oceans has promoted extensivere search on the topic [1,2].

Since redundant manipulators have more degrees of freedom than required for position and orientation, multiple solutions for this problem do exist. As a result, the redundancy of inverse kinematic mappings complicates the manipulator control problem considerably, in addition to its nonlinearity [3].

Generally, there are many algorithms for the redundant manipulator inverse kinematics. These algorithms can be divided into three categories [4], (1) the algebraic approach, (2) the iterative approach, such as neural networks, genetics algorithm, and fuzzy logic and (3) the geometric approach.

The algebraic approach suffers from the fact that it does not give a clear indication on how to select the correct solution from the several possible solutions for a particular arm configuration. The iteration solution does not guarantee convergence to the correct solution. Furthermore, just like the algebraic approach there is no indication of how to choose the correct solution for a particular arm configuration. The geometrical approach on the other hand can provide the solution directly with simple calculations.

In order to take full advantage of the redundancy, various computational schemes have been developed. Most current researchers have utilized the pseudoinverse technique. Because the inverse kinematics model gives an infinite number of solutions for redundant manipulator, consequently, secondary performance criteria can be optimized such as joint limits avoidance [5], singularity avoidance [6,7], and obstacle avoidance [8–10]. In this paper, the secondary performance criterion is avoiding the joint limits and the singular configurations. Singularities are a serious cause of difficulties in robotic analysis and control. Motions have to be carefully planned in the region of singularities. This is not only at the singularities themselves where there will be an unobtainable motion at the end-effector, but also in the region surrounding a singularity, the joint velocities will be extremely high, even for relatively small end-effector velocities. This can lead to numerical instability, and unexpected wild motions of the arm for small, desired end-effector motions. Interior singularity will not occur

* Corresponding author.
E-mail address: smryahya@yahoo.com (S. Yahya).

http://dx.doi.org/10.1016/j.neucom.2013.11.038
0925-2312

using the method in this paper, because the angles between the adjacent links are always the same. When two links are aligned this means that all other links will be aligned as well, and this configuration does not happen except when it is desirable to move the end-effector on the boundary The singularity and joint limits avoidance are discussed with details in the next section.

The kinematic equations describe the motion of the robot, without taking into consideration the forces and moments that produce the motion, while the dynamic equations explicitly describe the relationship between force and motion. The equations of motion are important when considering robotic designs, control algorithm design, and simulations [11].

Simulating the motion of the manipulator allows control strategies and motion planning techniques to be tested without the need to use a physically available system. The analysis of the dynamic model can help for the mathematical design of the arm prototype, and the computations of the torques required for the execution of typical motions provides useful information for designing joints, transmissions and actuators [12].

The two basic problems associated with the dynamics of robotic manipulators are the inverse and the forward problems. Because the inverse problem is purely algebraic, it is conceptually simpler to grasp than the forward problem, while also being computationally simpler [13].

Because the inverse dynamic model provides the joint torques and forces in terms of the joint positions, velocities and accelerations, it is described by [14]

$$\Gamma = f_1(\theta, \dot{\theta}, \ddot{\theta}) \tag{1}$$

with Γ is the vector of joint torques or forces and, θ, $\dot{\theta}$, $\ddot{\theta}$ are the vectors of joint positions, velocities and accelerations respectively.

The forward dynamic model describes the joint accelerations in terms of the joint positions, velocities and torques. It is represented by the relation [14]:

$$\ddot{\theta} = f_2(\theta, \dot{\theta}, \Gamma) \tag{2}$$

Many efforts have been made in studying the dynamics of robotic manipulators. Very few authors have studied seven degrees of freedom manipulator [15], or six degrees of freedom manipulator [16]. Other researchers have used six degrees of freedom, but due to computational complexities, they calculate the dynamics for just the first three degrees of freedom of the manipulator [17]. Some researchers use a simple three degrees of freedom planar manipulator [18], or even with only two degrees of freedom [19,20]. The reason is in the complexity of mechanisms models, which increases rapidly, in tandem with the number of degrees of freedom [21].

In the last few decades, artificial intelligent control using neural networks has undergone a rapid development to allow the design of a feedback controller for complex systems. Artificial intelligent control has proven to be very powerful techniques in the discipline of systems control, especially when the controlled system is difficult to mathematically model, or when the controlled system has large uncertainties and strong nonlinearities [22,23].

Artificial neural networks, especially Backpropagation network, have been successfully used in many fields of application. Recently, many supporting facts on Backpropagation networks have been demonstrated successfully in solving neural computation problems on linear algebra such as factorization of two-dimensional polynomials, inversing nonsingular matrices, solving linear simultaneous algebraic equations and zeroing polynomials. The advantage of these neural computation approaches is to provide more flexible structures and suitable adaptive learning algorithms for solving those linear algebraic problems. Generally, using Backpropagation networks, the corresponding solutions for the problems involved can be simultaneously obtained due to the parallel neural network structures. Specifically, when the data from the problems involved change, the corresponding neural computation approaches are capable of tracking the change by adaptive learning algorithms. So, it is envisaged that this novel neural computation approach is a vital research topic for intelligent computation field in the next decade [24–26].

Due to the fact that radial basis function neural networks provide a powerful technique for generating multivariate nonlinear mapping, and because of their simple topological structure, the learning algorithm corresponds to the solution of a linear problem, so the training of the network is rapid [27]. In this article, the radial basis function neural networks have been used to estimate the centrifugal and gravitational effects of the joints, while the end-effector is following a desired path.

To avoid or reduce the disadvantages of the radial basis function neural networks and the Probabilistic neural networks, the radial basis probabilistic neural networks was used in. The construction of radial basis probabilistic neural networks involves four different layers: one input layer, two hidden layers and one output layer. The first hidden layer is a nonlinear processing layer, which generally consists of hidden centers selected from a training samples set. The second hidden layer selectively sums the outputs of the first hidden layer, which generally has the same size as the output layer for a labeled pattern classification problem. In general, the weights between the first and the second hidden layer are set as fixed values (1 or 0) and do not require learning. Generally, the first hidden layer is tightly interrelated to the performance of the radial basis probabilistic neural networks [28].

Just as for the radial basis function neural networks, in the first hidden layer of the radial basis probabilistic neural networks, the hidden centers number and locations as well as the controlling parameters of the kernel function are quite important indices. Too many hidden centers will lead to very lengthy training and testing time, and poor generalization capability, while, too few hidden centers can lead to quite great convergent error. In addition, the selected hidden centers will require especial controlling parameters in order to realize the entire overlay of training samples in space [29].

Because of the radial basis function of neural networks ability, due to their fast training and convergence, it has been used in this work to calculate the simulation results. The next section compares the results of these two algorithms.

2. Kinematics of the manipulator

A manipulator with n joints, whose link position variables are denoted by $\theta = [\theta_1, \theta_2, \ldots, \theta_n]^T$, that are used to control m independent variables of the end-effector, can be described by $x = [x_1, x_2, \ldots, x_m]^T$ $(m \le 6)$, and is described by the following kinematic equation [2]:

$$\dot{x} = J\dot{\theta} \tag{3}$$

where $\dot{x}$ is an m-dimensional vector of Cartesian velocity (linear and angular) components of the end-effector with reference to the base coordinates, $\dot{\theta}$ is an n-dimensional vector of joint velocities, and J is an $m \times n$ Jacobian matrix. Unless otherwise stated, m will be assumed to equal to six, implying that six independent variables are required to describe the end-effector.

If J is a square matrix $(m = n)$ and has a rank equal to m (full rank), then joint velocities required to achieve the desired end-effector velocity will be unique and can be evaluated by

$$\dot{\theta} = J^{-1}\dot{x} \tag{4}$$

If J is singular or rectangular with $m < n$, the vector $\dot{\theta}$ can be computed by the following commonly used method:

$$\dot{\theta} = J^{+}\dot{x} + (I - J^{+}J)\dot{\phi} \tag{5}$$