Engineering Applications of Artificial Intelligence 74 (2018) 104–120

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

A soft computing approach for inverse kinematics of robot manipulators

Carlos Lopez-Franco *, Jesus Hernandez-Barragan, Alma Y. Alanis, Nancy Arana-Daniel [1]

University of Guadalajara, CUCEI, Computer Science Department, Mexico

ARTICLE INFO

Keywords:
Forward kinematics
Inverse kinematics
Soft computing
Path tracking

ABSTRACT

The solution of the inverse kinematics problem is an essential capability for robotic manipulators. This capability is used to solve tasks such as path planning, control of manipulators, object grasping, etc. In this paper, we present an approach for solving the inverse kinematics of robot arm manipulators using a soft computing approach. Given a desired end effector pose, the proposed approach is able to solve both the position and orientation for the inverse kinematic problem. In addition, the proposed approach avoids singularities configurations, since, it is based on the forward kinematics equations. We present simulations and experiments, where a comparative study among some selected soft computing algorithms is realized. The simulations and experiments illustrate the effectiveness of the proposed approach.

1. Introduction

The solution of the inverse kinematics of robot arm manipulators is a crucial task for solving path tracking and control problems. A robot arm can be consider a redundant manipulator when the robot exceeds the number of degrees of freedom (DOF) than strictly needed for executing a task. The manipulators with 6 or more DOF are able to reach any desired end effector pose, but they are always considered as redundant robots. The solution of the inverse kinematics of redundant manipulators may have as a result multiples joint configurations to reach the same end effector pose. These redundancy solutions may have singularities configurations and non linearities which become the inverse kinematic problem difficult to solve (Huang et al., 2012). In this paper, we propose an approach for solving the inverse kinematic problem of redundant robot manipulators.

Recently, soft computing techniques have become very popular, they have been successfully applied in many research and application areas. These algorithms have gained the attention of researchers in order to solve the inverse kinematics of redundant robots (Huang et al., 2012; Glumac and Kovacic, 2013; Nearchou, 1998; Yang et al., 2007; Li et al., 2012; Du and Wu, 2011; Pham et al., 2008; Tabandeh et al., 2006).

The inverse kinematic problem can be solved by minimizing an error function using an iterative process (Glumac and Kovacic, 2013; Sciavicco and Siciliano, 1996). However, conventional approaches suffers problems such as singularities configurations (Sciavicco and Siciliano, 1996). These singularities make the robot joint speeds necessary to achieve a configuration become excessive. In order to overcome such problems, we proposed the use of soft computing algorithms to solve the inverse kinematics, as an optimization problem. An objective function was formulated based on the kinematics of the redundant manipulator, where the position and orientation of the end effector are considered. During the iterative process, the proposed approach solves the inverse kinematics based on its forward kinematics equations, since the forward kinematics always have a solution to find a position and an orientation for the given joint settings. This approach avoids the problems of singularities configurations.

In this paper, we perform tests with different soft computing algorithms to solve the inverse kinematics problem for robot manipulators. The soft computing algorithms considered in this work are: Artificial Bee Colony (ABC), Bat Algorithm (BA), Covariance Matrix Adaptation Evolution Strategy (CMA-ES), Cuckoo Search (CS), Differential Evolution (DE), Differential Search (DS), Genetic Algorithm (GA) and Particle Swarm Optimization (PSO). A comparative study is presented, where all such method are compared solving the inverse kinematic and path tracking problems for some manipulators. We also compare this algorithms using nonparametric statistical tests such as The sign test and the Wilcoxon signed ranks test. Furthermore, we compare the proposed approach against a traditional inverse kinematics method. In addition, experimental results are presented, where the best soft computing algorithm is used for solving the path tracking problem in a real robot manipulator.

Then the contribution of this paper is an approach for the solution of the inverse kinematics problem for robot manipulators, including the

* Corresponding author.
E-mail address: carlos.lopez@cucei.udg.mx (C. Lopez-Franco).
[1] Since 1880.

https://doi.org/10.1016/j.engappai.2018.06.001
Received 10 June 2017; Received in revised form 4 May 2018; Accepted 4 June 2018
0952-1976/

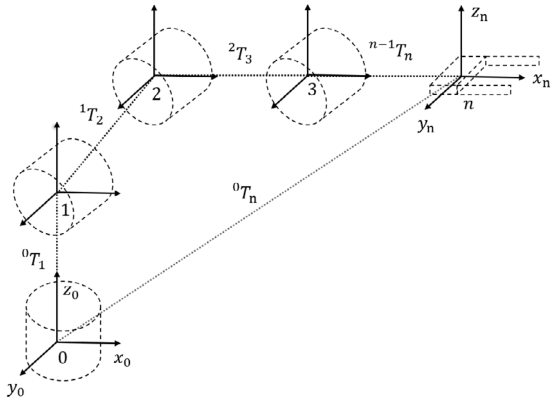

Fig. 1. Kinematic chain of a robot. $^{i-1}T_i(\theta_i)$ for $i \in 1,2,3,\ldots,n$ is the homogeneous matrix that represents the coordinate frame of the link i.

redundant robots arms. This approach solves the inverse kinematics problem for the given position and orientation of the end effector pose. Because the proposed approach does not require a Jacobian matrix and the forward kinematics always have a solution, there are not singular configurations as the result of the inverse kinematics. The simulation and experimental results prove the effectiveness of the proposed approach.

The rest of the paper is organized as follows: the next section will present a brief introduction to the forward and inverse kinematics. Then, in Section 3 the soft computing algorithms are described. In Section 4, the proposed approach is described. The results of simulations and real experiments are given in Section 5 and Section 6, respectively. Finally, in Section 7 we give the conclusions.

2. Problem statement

Robotic manipulators are composed of a series of rigid bodies (links) connected by joints, this structure forms a kinematic chain. The number of joints defines the degrees of freedom (DOF) of a robot manipulator. Each joint is associated with an articulation and it defines a joint variable. The computation of the end effector pose given the joint variables is a straight forward problem and it is called forward kinematics (Craig, 2005; Spong and Vidyasagar, 2008). On the other hand, the computation of the joint variables given the end effector pose is not straight forward and it is called inverse kinematics. The next subsection gives a brief introduction to the forward and inverse kinematics.

2.1. Forward and inverse kinematics

A robot system can be described from the kinematic model of its links and joint of the robot. Considering the robot's kinematic chain in Fig. 1, the forward kinematic can be calculated as

$$\begin{aligned} {}^0T_n &= {}^0T_1(\theta_1)\ {}^1T_2(\theta_2)\ {}^2T_3(\theta_3)\ \cdots\ {}^{n-1}T_n(\theta_n) \\ &= \prod_{i=1}^{n} {}^{i-1}T_i(\theta_i) \end{aligned} \tag{1}$$

where n is the total number of degrees of freedom, ^0T_n represents a homogeneous matrix that contains the position and orientation of the end effector pose and the parameters $\theta_i \in 1,2,3,\ldots,n$ represent the position for each joint of the robot.

The forward kinematics can be expressed as the product of the matrices that represents each link, from the beginning at the end. The kinematics of the robot can be described using the Denavit–Hartenberg (DH) model. In this model each link is represented by an homogeneous matrix $^{i-1}T_i(\theta_i)$ that transforms the frame attached to the link $i-1$ into the frame fixed to link i. This homogeneous matrix can be expressed as the product of four basic transformations

$$\begin{aligned} {}^{i-1}T_i(\theta_i) &= T_{rot_z}(\theta)\ T_{trans_x}(a)\ T_{trans_z}(d)\ T_{rot_x}(\alpha) \\ &= \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{2}$$

where the parameters θ_i, a_i, d_i and α_i are the parameters associated with the link and joint i. These parameters represent the joint angle, link length, link offset and link twist respectively. For brevity, we represent sin and cos operations with the letters s and c, respectively.

The homogeneous matrix $^{i-1}T_i(\theta_i)$ can be written as

$$\begin{aligned} {}^0T_n &= \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} [\mathbf{n} & \mathbf{o} & \mathbf{a}] & \mathbf{p} \\ & \mathbf{0} & & 1 \end{bmatrix} \\ &= \begin{bmatrix} R & \mathbf{p} \\ \mathbf{0} & 1 \end{bmatrix} \end{aligned} \tag{3}$$

where R is a rotation matrix that represents the orientation of the respectively frame i and the vector $\mathbf{p}$ represent its position.

The inverse kinematics consist in the computation of the joint variables given the pose of the end effector (Sciavicco and Siciliano, 1996). This can be defined as the computation of the vector $\mathbf{q}$ for the function (1), where, the vector $\mathbf{q}$ is defined as

$$\mathbf{q} = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 & \cdots & \theta_n \end{bmatrix}^T \tag{4}$$

For further information about forward and inverse kinematics, we encourage the reader to read (Sciavicco and Siciliano, 1996; Craig, 2005; Rafael Kelly Ph.D. and Davila Ph.D., 2005; Spong and Vidyasagar, 2008).

3. Soft computing algorithms

The solution of the inverse kinematics can be solve by minimizing an error function using some iteration process. This error function usually represents an euclidean distance between the desired and the actual end effector position. Given a 2 DOF manipulator with links length $L_1 = 0.5$ and $L_2 = 0.5$ meters, we can solve the inverse kinematics for a desired end effector position at the point $x = 0.5$ and $y = 0.5$ meters. In this case, there are two different arm configurations to reach the desired end effector position, see Fig. 2. One possible solution is given by $\theta_1 = 0$ and $\theta_2 = 1.5708$ and the other solution is given by $\theta_1 = 1.5708$ and $\theta_2 = -1.5708$ radians. Both solution are suitable if they are inside of the joints limits.

As we can see in Fig. 2, the inverse kinematics problem is given by a multimodal function. In this case, two local minimum represent the suitable joint configurations for the desired end effector position. However, many local minimum may be presented as the number of DOF increase, in consequence the optimization problem becomes challenging to solve. For this reason, it is necessary to search for no conventional methods to solve the inverse kinematics problem. In this work, we propose the use of soft computing algorithms in order to solve the inverse kinematics problem as a constrained optimization problem. The proposed approach is able to provide the optimal joints configuration for the given end effector pose. If there exist suitable multiples solutions for the inverse kinematics, the proposed approach provides whichever of these solutions. In the case of a path tracking problem, the solution of the inverse kinematics for every point in the path is computed taking into account the previous joints configurations. This last avoids abrupt changes in the joints configurations during the path tracking task.

Soft computing algorithms have been commonly used for solving many optimization problems such as classification, clustering, image processing and neural networks. Several soft computing algorithms have