Mechanism and Machine Theory 74 (2014) 310–318

Contents lists available at ScienceDirect

Mechanism and Machine Theory

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

Solving inverse kinematics by fully automated planar curves intersecting

Tomasz Rudny *

Faculty of Mathematics and Information Science, Warsaw University of Technology, Koszykowa 75, 00-662 Warsaw, Poland

ARTICLE INFO

Article history:
Received 3 October 2012
Received in revised form 22 December 2013
Accepted 28 December 2013
Available online 22 January 2014

Keywords:
Inverse kinematics
Serial manipulators
Planar curves
Bernstein elimination

ABSTRACT

This paper presents a new method for solving the Inverse Kinematics Problem of general 6-DOF serial manipulators. This problem has been a fundamental research area for the last 4 decades as it has numerous applications. Granted, many methods have been developed, but none of them is at the same time fast, accurate, numerically stable for an arbitrary end-effector pose and capable of finding all solutions to the problem. The inverse kinematics problem can be reduced to finding intersections between planar curves, as it was shown by Jorge Angeles. Here we present a new way of transforming those curves into bivariate polynomials without the application of half-tan substitution and then of finding their intersections numerically using Bernstein elimination. It is achieved without any human interaction, so the method is fully automated, thus suitable for applications.

1. Introduction

1.1. Background

Serial manipulators play an important part in applications — NC milling, robot control, computer graphics and animation. Kinematic chains – the mathematical notation of serial manipulators – can be used not only in industry, but also to describe various objects of nature e.g. skeletons. Manipulators consist of links which are connected with joints. Here, we consider joints of only two types — revolute and prismatic, allowing to rotate or slide correspondingly a link against a given axis. All other types of joints can be described as a composition of (possibly many) joints of those two simple types.

Following a Denavit–Hartenberg (D–H) notation, each joint can be described by four parameters. Their values fully determine the transformation needed to transform a local Cartesian coordinate frame associated with the link L_i into the frame associated with the next link L_{i+1}. For a revolute joint one D–H parameter, the angle θ_i is the joint variable, while for a prismatic joint it is the distance between some joints axes — d_i.

When the joints variables are given for a kinematic chain, then the pose of its last link, the end-effector can be easily found by simple matrix multiplication. However, the inverse is not true: finding all joints variables when knowing the pose of the end-effector is a difficult computational problem.

Practical applications require that a method for solving the inverse kinematics problem is:

1 capable of finding all solutions to the IKP for the given pose of the end effector,
2 efficient i.e. with solve time acceptable in CAD/CAM and possibly in robot control,
3 accurate i.e. the error of the found solutions approximations should not be greater than 0.001 rad or 0.001 mm,
4 numerically stable i.e. should converge for any pose of the end effector (or return error message for singular configurations).

* Tel./fax: +48 61 278 64 18.
E-mail address: rudnyt@mini.pw.edu.pl.

0094-114X/$ – see front matter © 2014 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.mechmachtheory.2013.12.020

In this paper we present such method.

1.2. Existing methods

Inverse kinematics problem has been studied by many great researches for the last 40 years. The first discoveries were made by Pieper ([13]) and later by Freudenstein. In 1972 Roth et al. proved that a 6R open manipulator can admit at most 32 poses. In 1980 Duffy and Crane found a way to transform the IKP into the problem of finding roots of a polynomial of degree 32. In 1985 Tsai proved that for a general 6R manipulator the number of solution is at most 16. In 1989 Raghavan and Roth found a polynomial of degree 16 describing the IKP [14]. In 1992 Manocha and Canny ([9]) modified this method to look for eigenvalues of a companion matrix instead, which improved stability and accuracy of the method.

Since then many methods have been designed and tested, including optimization methods ([16,19]), neural networks ([8]), interval algebra ([4]), genetic algorithms ([17]) to name just a few. Many of the new methods focus on deriving the polynomials in a more efficient way or one that reveals the geometric structure of the problem ([7]).

Jorge Angeles noticed that instead of deriving a univariate polynomial, a set of 4 bivariate equations in variables θ_4, θ_5 may be obtained. These define 4 contours on the $\theta_4 - \theta_5$ plane.[1] The common intersection points of these correspond to the solutions of the IKP. Angeles developed a GUI in Matlab that allowed plotting these contours. A user would then localize intersections with a mouse. These approximations were then improved using a Newton method.

A comprehensive discussion of planar curves in computations can be found in [6]. The comparison of the efficiency of different implicit curve plotting methods based on interval arithmetics, Bernstein coefficient, continuation and others is given in [10]. These methods can be also applied to curves intersecting. However, none of these general methods provide needed accuracy within the needed time.

One key observation is that curves intersecting is much faster for polynomial curves.

2. Inverse kinematics as planar curves intersecting

2.1. Transforming equations

Let the elements of the matrix describing the pose of the end-effector be given by:

$$\mathbf{A}_{hand} = \begin{bmatrix} l_x & m_x & n_x & p_x \\ l_y & m_y & n_y & p_y \\ l_z & m_z & n_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{1}$$

This matrix subjects to the matrix equation:

$$\mathbf{A}_1\mathbf{A}_2\mathbf{A}_3\mathbf{A}_4\mathbf{A}_5\mathbf{A}_6 = \mathbf{A}_{hand} \tag{2}$$

or equivalently:

$$\mathbf{A}_3\mathbf{A}_4\mathbf{A}_5 = \mathbf{A}_2^{-1}\mathbf{A}_1^{-1}\mathbf{A}_{hand}\mathbf{A}_6^{-1}. \tag{3}$$

Now taking the 3rd and 4th columns of the matrix Eq. (3) we can write new equations ([14,1]):

$$\overline{\mathbf{f}} = \overline{\mathbf{g}} \tag{4}$$

$$\overline{\mathbf{h}} = \overline{\mathbf{i}} \tag{5}$$

where:

$$\overline{\mathbf{f}} \equiv \begin{bmatrix} 1 & 0 & 0(6) \\ 0 & -\lambda_2 & \mu_2(7) \\ 0 & \mu_2 & \lambda_2 \end{bmatrix} \left(\begin{bmatrix} \mathbf{c}_3 & \mathbf{s}_3 & 0(8) \\ \mathbf{s}_3 & -\mathbf{c}_3 & 0(9) \\ 0 & 0 & 1 \end{bmatrix} \widetilde{\mathbf{f}} + \begin{bmatrix} \mathbf{a}_2(10) \\ \mathbf{d}_2\mu_2(11) \\ \mathbf{d}_2\lambda_2 \end{bmatrix} \right) \tag{6}$$

$$\overline{\mathbf{g}} \equiv \begin{bmatrix} \mathbf{c}_2 & \mathbf{s}_2 & 0(13) \\ \mathbf{s}_2 & -\mathbf{c}_2 & 0(14) \\ 0 & 0 & 1 \end{bmatrix} \widetilde{\mathbf{h}} \tag{7}$$

$$\overline{\mathbf{h}} \equiv \begin{bmatrix} 1 & 0 & 0(16) \\ 0 & -\lambda_2 & \mu_2(17) \\ 0 & \mu_2 & \lambda_2 \end{bmatrix} \begin{bmatrix} \mathbf{c}_3 & \mathbf{s}_3 & 0(18) \\ \mathbf{s}_3 & -\mathbf{c}_3 & 0(19) \\ 0 & 0 & 1 \end{bmatrix} \widetilde{\mathbf{r}} \tag{8}$$