Is it Necessary to Solve the Redundancy Problem when Learning the Inverse Kinematics of a Robotic Arm?

Rasmus Bååth
rasmus.baath@lucs.lu.se

Christian Balkenius
christian.balkenius@lucs.lu.se

Lund University Cognitive Science
Kungshuset, Lundagård
S-222 22 Lund, Sweden

Abstract

The aim of this study was to investigate if it is possible to learn the inverse kinematics of a redundant arm without explicitly solving the redundancy problem. A 2D two-joint arm was simulated and a feed forward artificial neural network was trained to associate a desired endpoint and a current joint angle configuration with a correct joint angle displacement. Training examples were collected by allowing the arm to move around randomly. The model performed well, correctly reaching the desired endpoint 96% of the time. This shows that a system capable of learning inverse kinematics do not need to explicitly solve the redundancy problem given the right kind of training examples.

1. Introduction

A problem one faces when building a robot with arm-like actuators is the inverse kinematics (IK) problem, that is, given a target point in space what joint angle configuration positions the arm endpoint at that point? If the configuration of the robots actuators are known an exact solution could be supplied by the constructor, but a more flexible solution would be if the robot could learn the IK. One problem faced by both humans and robots is that arms often have redundant degrees of freedom. Redundancy is useful as it increases the flexibility of the arm, but it also poses a problem in that there are more than one correct way to position the hand at an endpoint. The redundancy problem is then: What path should be taken out of the many possible that results in the hand reaching the endpoint?

Earlier attempts to learn to solve the IK problem have used one out of two strategies. Strategy one is to use a non-redundant arm (see e.g. Van Der Smagt and Krose, 1991) which reduces the problem to a mapping problem which can be solved by for example a standard feed forward artificial neural network (ANN). The other strategy (used by e.g. Butz et al., 2007) is to add an extra mechanism that explicitly deals with the problem of redundancy.

Here it is investigated whether learning the IK of a redundant arm can be done using a standard learning algorithm without explicitly solving the problem of redundancy. The idea is to generate training examples of small movements by allowing an arm to move around in a random fashion. When small movements have been learned, training examples with larger movements are generated. This is inspired by the “motor babbling” of infants and earlier successful applications of this to learning robot systems (Demiris and Dearden, 2005).

2. Method

A 2D two-joint arm was simulated. Each limb had a length of $1/4$ and each joint was allowed to bend in any angle from $-180^\circ$ to $180^\circ$. The base of the arm was placed at coordinate $(0.5, 0.5)$. This allowed the arm endpoint to reach any point within a circle with diameter 1.0 centered at $(0.5, 0.5)$. Note that the arm configuration is redundant, every arm endpoint can be reached by two different joint angle configurations. All simulations were done using the MATLAB programming language.

A feed forward ANN was trained to associate a desired endpoint, $G = (x_g, y_g)$, and the current joint angle configuration, $\theta = (\theta_1, \theta_2)$, with the joint angle displacement, $\theta' = (\theta'_1, \theta'_2)$, that positions the arm endpoint at the desired endpoint. The ANN had one hidden level with 20 neurons, the input and hidden layer used a tan-sigmoid transfer function and the output layer used a linear function. The training function used was Levenberg-Marquardt backpropagation.

Training examples were collected by allowing the arm to “motor babble”. The arm was initially placed using a randomized $\theta$. Then $\theta'$ was assigned a normally distributed random number with $\mu = 0$ and $\sigma = 8^\circ$. $G$ was given by reading he arm endpoint after moving the arm by $\theta'$. This random joint dis-
placement was repeated until \( n \) training examples had been collected, these were then given to the ANN to train on for five epochs. This cycle of generating training examples was repeated three times with successively larger \( \sigma \), \((10^\circ, 12^\circ, 15^\circ)\).

To evaluate the performance of the trained ANN it was measured how long the path of the arm was when reaching for a randomly placed \( G \). The path length was measured as the sum of the angle distance the two joint traveled to get to \( G \). Given \( G \) and \( \theta \) the ANN generated a predicted \( \theta' \) and the arm was moved accordingly. Using the new \( \theta \) a new \( \theta' \) was generated and the arm was moved again. This continued until \( G \) was reached. If 50 moves had been made without the arm reaching \( G \) the reach was not counted and labeled as “failed”. For each trained ANN 100 \( G \) were generated in total.

To investigate how the number of training examples, \( n \), influenced the performance, ANNs where trained using \( n = (2^6, 2^7, \ldots, 2^{14}) \). For each \( n \) 10 ANNs were trained and used in the evaluation.

3. Result and Discussion

Figure 1 shows the mean path difference between the predicted reaches and optimal reaches. Optimal in this case means a reach movement that minimizes the joint angle displacement. The minimum difference is found at \( n = 2^{14} \) and is 54°. Figure 2 shows the mean number of failed reaching attempts while making 100 successful attempts. The minimum is found at \( n = 2^{14} \) and is 4.1.

The aim of the current study was not to propose a new model for learning IK but to investigate if it is possible to learn the IK of a redundant arm without explicitly solving the redundancy problem. Given this aim the model performs surprisingly well. When the number of training examples are large the arm manages to reach the desired endpoint 96% of the time. The mean difference between the predicted path and the optimal path is also quite low, 54°.

A limit of the model is that presented with a goal position only one solution is given.

The question posed in the introduction was: Is it necessary to explicitly solve the redundancy problem when learning the inverse kinematics of a robotic arm? Given the result of the experiment presented the answer would be "no". While earlier attempts have solved the problem of learning IK by explicitly dealing with the problem of redundancy, the presented model relies on the format of the training data to render the problem of redundancy a non-problem. Thus when constructing a learning robot one might not need a redundancy solving component and the search for such in the human motor system, might not be necessary.

Acknowledgements

The authors would like to thank the Linnaeus Center for Cognition Communication and Learning for its support.

References

