Intelligent adaptive systems coursework 1

# Relevant examples of IK learning

Various different approaches have been used in inverse kinematics learning such as using Anfis, neural networks and genetic algorithms. By far the most popular are artificial neural networks, these can be broken down into various types such as multilayer perceptron’s, radial basis functions and recurrent neural networks.

One approach taken by ANfis RRR paper used Anfis to learn the inverse kinematics of both a 2 joint and 3 joint planar manipulators. The number of Anfis system was the same as the number of joints, the inputs into the 2 joint system were the X and Y end effector locations. While the input into the 3 joint system was the X, Y and Phi. The paper states that they achieved a suitable level of error. The approach used here will be very similar to the one used for this coursework. This paper does not describe how many membership functions were used.

Another study by Second anfis paper also used a similar approach, this study solely focused on a 3-joint manipulator. The inputs were the X Y and Phi (EE pose) of the end effector position corresponding to the thetas. It use 3 Anfis systems with 6 membership functions each and each of these were trained with 100 samples and 200 epochs; it matched the chosen trajectories quite well. Photo?

One study by perceptron1 looks at using a multi-layer perceptron for inverse kinematics, the inputs to the network are the X, Y and phi. The chosen structure had one hidden layer of 100 sigmoid activated neurons with 3 outputs neurons corresponding to the joint angles. The network can match the desired output arbitrary well. This study notes that further improvements to the network could be made, such as looking at the size and number of hidden layers. This paper is used as a reference for the MLP developed in this assignment. Photo

A study by RBF and ANN looked at both MLPs and RBFs for inverse kinematics of a 3R planar manipulator and compared them. The inputs to both types of network were the X, Y and Phi. The paper describes details of both of the systems developed such as the learning used.

Some studies such as the one by MIXED have looked at combining approaches to reduce error. In this approach the inverse kinematics is calculated with 3 separate systems (2 MLP’s and one RBF) and the error of each is compared and the best one is chosen.

* Genetic algorithms, any other approaches?
* Small comparison summary?

# Inverse kinematics with Anfis

## Data/workspace generation

In order to generate a dataset that Anfis could be trained on a workspace was generated using the forward kinematics equations for a 3R planar manipulator with link lengths of 10,7,5 for l1,l2 and l3 respectively. These are:

* FK equations

These positions were calculated across a range of joint angles. The joint limits for each joint are from 0 to pi for theta 1, from 0 to pi/2 for theta 2 and from -pi/2 to pi/2 for theta 3. Note that problems may arise due to the possibility of theta 3 being either positive or negative, which will result in locations with multiple solutions. The size of the dataset can be adjusted by changing the interval in the for loop (this can be done for a specific angle or for all). The dataset for the pose (Phi) is also created at this time, as it is the summation of the joint angles.

The end result is a dataset with a variety of end effector poses (phi’s), positions and their associated joint angles which will be the training output. This is similar to the approach used in ANFIS RRR PAPER, the workspace is shown in workspace figure.

* Workspace photo here

## Anfis structure & training

* GENFIS for initial design? Chosen membership function type?

For learning inverse kinematics, a set of 3 Anfis systems were used, the inputs to each were the X & Y of the generated end effector locations as well as the corresponding Phi(pose), the training outputs of each system were the corresponding joint angle (theta), this is similar to the approach taken by ANFIS RRR PAPER.

The number of membership functions for each of the systems in the study by ANFIS 2 was 6 for each, these were the starting parameters for this assignment. The number of epochs was chosen for each of the Anfis systems and they are for the first, for the second and for the third.

These values were selected by testing which number of membership functions gave the most accurate inverse kinematics and then slowly adjusting the number of epochs until suitable generalization occurred. HOW DID WE GET THERE?

* Relate this to the relevant paper?

## Validation

The chosen error metric for validation is the error as a percentage of the reach radius (Which is 22 units). In order to validate the system, several XY positions were input into the system and the theta that is output was then fed into the forwards kinematics equations and then plotted. The calculated end effector position by Anfis was then compared to the desired position. It was tested over a range of XY locations as well as poses to ensure good generalization. These are shown in figure.

* Show Anfis IK path on workspace? Multiple paths please

The inverse kinematics of the manipulator were also calculated analytically, this enabled a comparison between the output joint angles and positions of both systems. The overall error of the system is suitably low as shown by figure which shows the error across each of the chosen XY validation points.

* IK equations?
* Comparison to actual IK calculations? Error graph
* Asking for an actually possible configuration!

## Discussion

* Discuss general accuracy and ease of this method?
* Benefits of using Anfis?
* 2nd joint is restricted to only be elbow up/elbow down?
* Problems with 3rd joint? Has trouble with points it could reach with either configuration?

# Inverse kinematics with MLP

## Data/workspace generation

In comparison to Anfis multi-layer perceptron’s can work through a larger dataset. The dataset was generated in a similar fashion, however the intervals were much smaller resulting in a much denser workspace with a larger dataset as a result, as shown in figure.

* Picture of workspace

## MLP structure & training

The study by perceptron1 used one hidden layer with 100 nodes for IK learning. However, it is noted in the study that further work involving optimization of the network is necessary. As such the starting point for this assignment was with these parameters.

* Map any function with two layers?

The final MLP structure had hidden layers with 10 then 10 nodes. The chosen training method is Backprop, as this…

The generated dataset was partitioned into training, validation and testing, the ratio was 0.6:0.2:0.2 respectively.

* Number of epochs?
* Activation function?
* Gradient/error?

## Validation

* Talk about the actual MLP performance as well as your own validation set
* Performance graph
* Comparison of IK’s, error graph
* Same error metric
* Show IK path on workspace?

## Discussion

* Compare to Anfis, can use a much larger data set, more accurate
* MLP has trouble with the edges of the workspace. This is due to the training dataset containing less pose configurations for end effector positions on the edges of the workspace (These are still within the dextrous workspace; this is confirmed by analytical inverse kinematics), these tend to be in areas where the workspace is less dense which is shown in figure. However, this can be offset by having a larger dataset but that results in a longer training time.

# Problems with IK learning

* Problem with elbow up or down
* Reducing joint velocity as part of a cost function?
* How was this solved?
* Implementation?

# Search algorithms

* Genetic algorithms
* Implementation?

# REFERENCES

# APPENDIX - CODE