Gesture Recognition Using Evolution Strategy Neural Network

Johan Hägg
Intelligent Sensor Systems
Mälardalen University
P.O. Box 883 Västerås, Sweden
jhg05002@student.mdh.se

Batu Akan
Intelligent Sensor Systems
Mälardalen University
P.O. Box 883 Västerås, Sweden
batu.akan@mdh.se

Baran Çürüklü
Intelligent Sensor Systems
Mälardalen University
P.O. Box 883 Västerås, Sweden
baran.curuklu@mdh.se

Lars Asplund
Intelligent Sensor Systems
Mälardalen University
P.O. Box 883 Västerås, Sweden
lars.asplund@mdh.se

Abstract

A new approach to interact with an industrial robot using hand gestures is presented. System proposed here can learn a first time user's hand gestures rapidly. This improves product usability and acceptability. Artificial neural networks trained with the evolution strategy technique are found to be suited for this problem. The gesture recognition system is an integrated part of a larger project for addressing intelligent human-robot interaction using a novel multi-modal paradigm. The goal of the overall project is to address complexity issues related to robot programming by providing a multi-modal user friendly interacting system that can be used by SMEs.

1. Introduction

Introduction of industrial robots in the manufacturing process has revolutionized a number of industries. The most striking example is perhaps the car industry. An investment involving industrial robots or in automation in general is, thus, seen as a necessary action for strengthening market position of a company. Despite this strong belief robot automation investments are considered to be technically challenging as well as costly by a considerable amount of small and medium sized enterprises (SME) [6].

Some of the reasons for this belief are as follows: An industrial robot must be placed in a cell that will occupy valuable space. The robot will perhaps operate only during a couple of hours a day. In addition, no matter how simple the manufacturing process might be, to integrate the robot into a manufacturing process one has to rely on a robot programming expert, a so-called integrator. Robot programming requires expert knowledge not only in robot kinematics, the integrator must also master advanced software engineering. For an engineer that does not have skills of a robot integrator switching from one manufacturing process to another is not a trivial task. The integrator is, thus, needed when there is a need for a new or updated robot program. Obviously, for most SMEs these issues result in challenges with respect to high costs, lack of flexibility, and reduced productivity. Note that these challenges are not restricted to SMEs. There is always a need of increased flexibility and optimally usage of resources in all sizes of companies as well as sectors.

We hypothesize that in order to make industrial robots more common within the SME sector, industrial robots should be (re)programmable by task experts as well. In this context we define a task expert as a person that has expert knowledge on the manufacturing process that is subject to an industrial robot investment. As describe above, no matter how knowledgeable the task expert is, he/she will need assistance while integrating the robot into the manufacturing process. Yet another important factor in this context is related to how a novel user is introduced to a totally new way of interacting with a robot. The importance of this factor should not be underestimated since getting used to a new system is always associated with new challenges for an individual.

In this paper we present a hand gesture recognition system. This system is part of a larger system called the µ-Intelligent Human-Robot Interaction Architecture (µ-iHRI) (Fig. 1). It consists of a novel high level language and a cognitive robot architecture (µ-for addressing natural interaction between a human and a robot [3] (see [4] and [5] for an introduction to the field). The µ-iHRI system is designed to be highly intuitive to use for task experts as well as other user groups. This system will allow the user to instruct the robot, a process which is fundamentally different from traditional robot programming. This process will be carried out using combination of following modalities: (i) hand gestures, (ii) natural speech, vision-based 3-dimensional object recognition, and (iv) vision-based hand and body posture recognition. Note that these modalities are used naturally by humans while interacting with each other. In the µ-iHRI paradigm a robot will be able to give feedback to the user when needed. This is highly important since all communication, including ambiguities and other complex issues, must be dealt within the framework of the same framework.

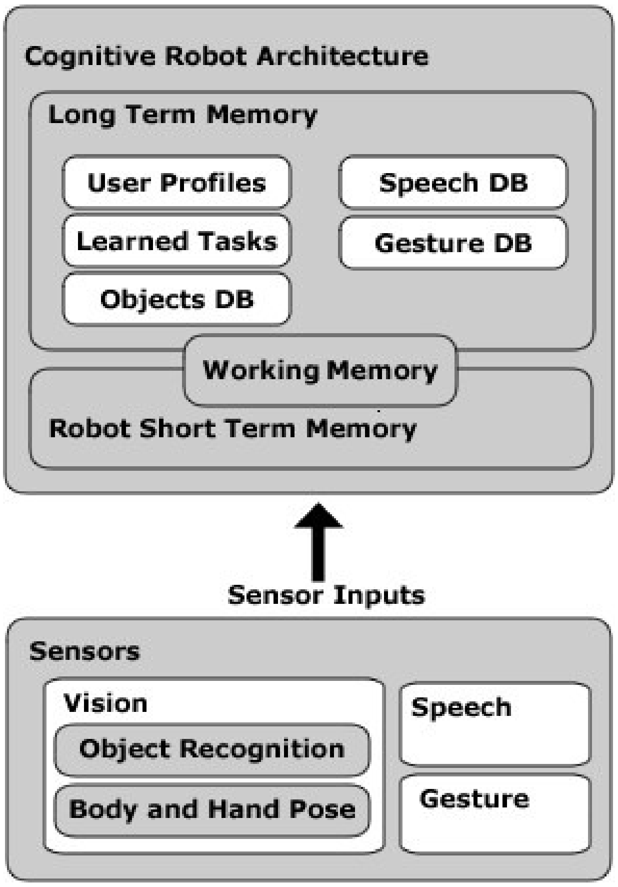

Figure 1. The µ-Intelligent Human-Robot Interaction Architecture.

Consequently, the high level language is equipped with methods for representing objects, their features, and relationships between them so that meaningful instructions can be given to the robot. Taking together, it is obvious that the process of interaction, as described above, is more like a dialog between two humans than traditional robot programming, and the gesture recognition system under development is an integrative part of this larger system.

2. Methods

Gesture recognition has to be addressed with an algorithm that detects gestures with high probability, since the system has to be reliable. A system that does not meet this requirement will not be accepted by users in general. Furthermore, a new user should spend as little time as possible with the system during the introduction phase, which also includes teaching the gesture recognition system the new user's gestures. It is not trivial to address both these requirements at the same time, since to fulfill the first requirement, usually, one expects to collect considerable amount of data from a user. This can be interpreted as a time consuming process, and hence may not be plausible in many cases. Thus, high system performance and user friendliness are clearly in contradiction with each other.

Yet another requirement is on the reaction time of the algorithm, since unnecessary long delays are considered to be annoying, especially when the user expects real time performance from the system. These three requirements are directly related to usability of gesture recognition systems as well as the whole system, namely the µ-iHRI, and hence have to be addressed.

Feed-forward artificial neural networks (ANN) [1] have been used in a number of function approximation problems with success. An ANN is an abstract topology consisting solely of computing units called perceptrons, which are grouped into layers. Information is propagated in a poorly feed-forward manner from input to output layer without feedback to previous layers. Connections between perceptrons in a layer are also prohibited. Each of these connections is associated with a weight, which is adjusted by a learning algorithm during the learning phase. The result is approximation of the target function. Thus, input to a perceptron is weighted sum of the outputs of perceptrons from the previous layer. Output of a perceptron takes the form of a sigmoidal transfer function [1]. In the proposed work ANN are trained using the evolution strategy (ES) paradigm [2]. This class of algorithms searches the domain in a probabilistic way. They are suitable for noisy data and search domains having many local maxima points.

The gesture data is collected by a Senseboard device [3]. This U-shaped device is attached to a hand, parallel to the wrist. It is fitted between the thumb and rest of the fingers. The Senseboard is equipped with five sensors: three accelerometers, one for each dimension (x,y,z), and two tilt sensors (wrist up/down and elbow rotation). Data from the Senseboard is, thus, five time series. Once data collection is done for a gesture, it is manually tagged with the gesture type information and number of samplings.

ES is a probabilistic population based search algorithm. It works by randomly selecting individuals from a generation and produce new individuals, called offsprings, by performing crossover on the selected individuals. Later, the offsprings are randomly mutated. Once this has been done, the fitness of each offspring is calculated. The fittest individuals from the intermediate population consisting of offsprings and parents are selected into the next generation.

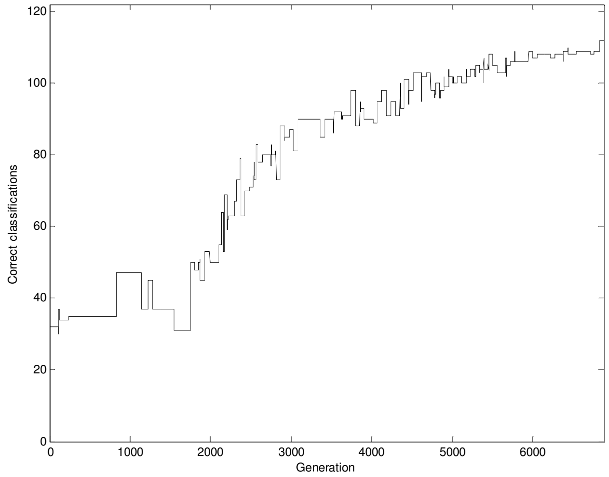

Figure 2. Number of correctly classified gestures as a function of generation. After 7000 generations the ANN can detect almost 80% of the gestures.

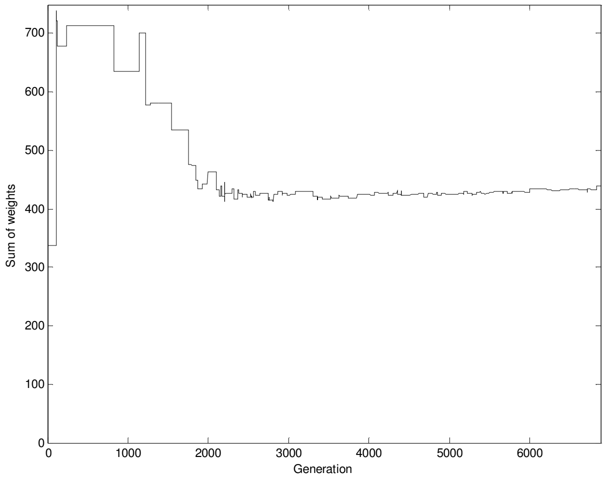

Figure 3. Sum of absolute weights as a function of generation. The penalty function for the weights prevents them from grooving in an uncontrolled manner.

Yet another advantage of using ES instead of traditional methods for training ANN, such as back-propagation, is that the network topology can be optimized during training, with respect to number of connections. Since gesture data is high dimensional this possibility is seen as a way of improving results even further in the future. Note also that adding new gestures into the language will also induce more complexity, thus the need for removing connections as well as perceptrons will probably needed.

3. Results and Conclusions

Collected data consists of four different types of gestures, defined as “rotate up”, “rotate down”, “rotate left”, and “rotate right”. In addition a sampling series with no hand movement is also performed. These data are tagged as “no-gesture”. All gestures are truncated to become exactly one second in length (25 sample points with 25 Hz sampling frequency). In total there are 142 gestures evenly distributed between five gesture types. Note that there are only 28–30 gestures representing each gesture type. This is for testing the network performance with as few gestures as possible to see if the proposed algorithm can address the second requirement defined earlier.

The training data is constructed by randomly dividing all 142 gestures into five sets. The algorithm performing this procedure ensures that all five gestures types are distributed as evenly as possible. Later, the dimension of the data is reduced to one fifth of its original by calculating mean values of five adjacent data points along the time axis. This process reduces noise is the data as well as dimension of it. Finally data from all five sensors are combined into a vector of 25 elements, which will become the input to the ANN. The ANN topology used for function approximation consists, thus, of 25 inputs. It has two hidden layers of perceptrons, eight in the first, and six in the second. The output layer has four perceptrons. The resulting network topology contains 290 weights. The initial value for each weight and bias value is 0.0±2.4. In the training data the output that is associated with a certain gesture is set to 1.0, whereas others three outputs are set to 0.0. All four outputs for “no-gesture” data are set to 0.0.

Population size is 80, and hence is constant. Each offspring has two parents, and in each generation 10 new offsprings are generated. It should be noted that the mutation used in ES is a normal distributed random number with the current value as its origin, the individuals’ mutation strength is the standard deviation of this distribution.

One of the optimization methods available in ES, self-adaptation, has been proved to be valuable for the gesture data. This method works by keeping track of how many offspring outperform the parents. This value is then used at regular intervals to adjust the mutation strength of each individual. This value is the number of successful offspring divided by the number of offspring produced and is called probability of success. If this value exceeds a certain threshold, mutation strength is increased, if it is lower than the threshold mutation strength is decreased, otherwise nothing is done. In the simulations mutation strength for each individual is set to 4.0±1.0% at the beginning. The self-adaptation algorithm controls every thirty generations with a probability of success threshold of 1/5 and adjusts the mutation strength of each individual by 3%.

During training a penalty value equal to the sum of absolute weights divided by 100 is subtracted from the fitness value. The effect is reduced absolute weight values. This has three favorable affects. Firstly, connections between neurons as well as perceptrons can be removed, after the training phase. The result is simpler ANN, which are most often more favorable over their more complex counterparts. Furthermore, if connections from a certain input note are close to zero that node can be removed, and hence the ANN input dimension can be reduced.

During the first 100 generations of training, all five sets are used for training. Later, 5-fold cross-validation is used for deciding when to stop training the ANN. In this version of cross-validation one (out of five) sets is randomly picked and used as the validation set during 10 generations. Remaining four sets are used as training sets. The training procedure is repeated until one of the following convergence criteria is met after the first 100 generations: (i) 10 000 generations have passed; (ii) sum of squared errors over test set increases by more than ten during one generation; (iii) mutation strength of fittest individual is less than one.

Once the four-dimensional output matrix has been produced by the ANN, it can be used to calculate number of correctly identified gestures. The output that has the highest value, out of four, is the classified gesture. If the highest output value is less than 0.3, the gesture is classified as "no-gesture". Note that, once the network has been fully trained and is used in an application, the threshold for classifying the "no-gesture" can be modified for improved performance.

A number of experiments have been done to fully explore results of training ANN with ES for gesture data. ANN with one and two hidden layers have been tested, as well as the impact of self-adaptation on search performance. In general ANN with two hidden layers performed better, thus, for the final tests this topology was chosen.

Self-adaptation was surprisingly useful. With as few as 142 gestures in the training and validation sets ANN can detect roughly 80% of the data. In Fig. 2 the network converges to 112 correctly classified gestures out of 142. When self-adaptation was turned off correct classification was roughly 40% (not shown here). One of the main requirements of this system was minimum time spent to introduce a new user to the system. We hypothesize that for a gesture language consisting of some ten gestures it will not be plausible to do more than 30 runs per gesture. Note that only this procedure would take some 10 minutes. Finally, penalty function for the weights has shown to be useful (Fig. 3). Despite improved classification after 2000 generations sum of absolute weights is kept almost constant. In the next phase weights that have low amplitudes are subject to elimination. The result will be a more simpler network topology.

In the next phase of the project we intend to improve the classification performance further. We believe that 80% of correctness in classification may not be satisfactory for some users. We also need to address a richer gesture language, with perhaps four times as many gestures as in the previous language.

References

[1] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice Hall, 1998.

[2] Z. Michalewicz, *Genetic Algorithms +Data Structures = Evolution Programs*, Springer, 1998.

[3] B. Akan, A. B. Çürüklü, L. Asplund, "Interacting with Industrial Robots through a Novel Multi-Modal Language", *To appear in the proceedings of the International Symposium of Robotics*, 2008.

[4] M. A. Goodrich, A. C. Schultz, "Human-Robot Interaction: A Survey", *Foundations and Trends in Human-Robot Interaction*, Vol.1, pp.203-275, 2007.

[5] G. Biggs, B. MacDonald, "A Survey of Robot Programming Systems", *Technical report, 2003.*

[6] R. D. Schraft, C. Meyer, "The need for an intuitive teaching method for small and medium enterprises", *Technical report*, 2006.