Soft Computing in Excavator Application

Heikki Palomäki
Seinäjoki University of Applied Sciences
School of Technology
Seinäjoki, Finland
heikki.palomaki@seamk.fi

***Abstract*—Artificial intelligence seems to be a demanded method, suitable for realize only by the high-power computers and far from the everyday living. The one part of Artificial Intelligence is Soft Computing. These new calculation methods ware found by following for example the ants, the bees and the functions of the neural shells. Soft Computing includes for example neural network, fuzzy logic, genetic algorithms and swarm intelligence.**

Soft Computing is suitable for the practical applications, also. As an example, it can calibrate automatic the excavator deep meter system. The neural network can replace the demanding manual calibration of the tilt sensors. In addition, the neural network can find the length of the excavator bars automatic. While testing the calibration by the simulation, the promising results were found. Therefore one student team tested this method in practice with a small excavator model and found the same results.

Keywords — soft computing; neural network; excavator; simulation; calibration; acceleration sensor

I. INTRODUCTION

The use of the theoretical mathematic methods are not much used in customary practical applications. Especially artificial intelligence applications are realized using high-power computers. Although, soft computing can be applied in very simply way in individual practical subjects. The best known method is the neural network. The principle of one neuron in network is very simple and logical. More complicated neural networks are only a combination of single neurons.

In this contribution, the aim is to apply neural network into an excavator to calibrate tilt sensors and bar lengths. The excavator needs the accurate position of the bucket tip as preparing the base of the building foundation or digging the diversion ditch. The accuracy depends on the measured dimensions of the bars and the accuracy of the tilt sensors. The installation of the measuring system requires care as assembling the sensors and measuring bar lengths. The different excavating work phases uses different bucket form, length and the sensor angle. This all changes the depth calculating parameters. This means much work and many error possibilities in manual calibration. The idea of the automatic tuning method is, that let the excavator measure and calibrate itself. It already has sensors and with neural computing, this kind of automatic tuning method seems to be possible.

The content of this article is based on source [1].

II. SOFT COMPUTING

Artificial intelligence (AI) is a large concept copying the information processing methods from nature: spoken languages, nervous system, insect swarm, plant and animal tissue etc. The most technical concept of AI is soft computing (SC). It includes machine learning, neural networks, fuzzy logic, genetic algorithm, swarm intelligence and many other computing methods. One concept is shown in Figure 1.

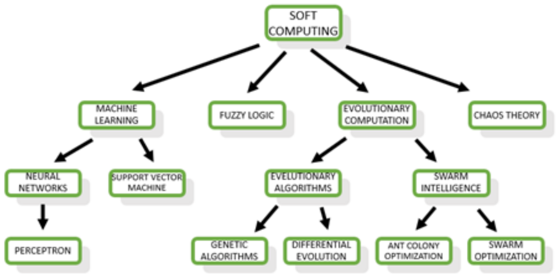


Figure 1: Soft computing concept

A. *Fuzzy logic*

A traditional proportional-integral-derivative (PID) controller is suitable for controlling linear processes and systems. If the process behaves non-linear, it includes many near-linear slices. These slices need different individual controller rules (Figure 2). At the borders of the slices, the controller can use two different rules alternately causing instability in control. Fuzzy logic means that borders are not exact. Control can be a combination of two rules weighted by the nearness of the corresponding rule as shown in Figure 3.

Reference [2] includes more material about fuzzy logic.

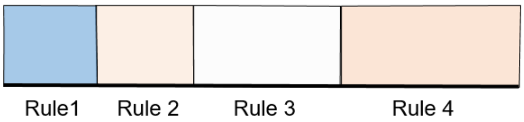


Figure 2: Traditional non-linear control

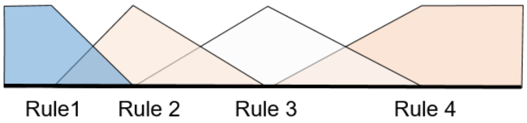


Figure 3: fuzzy non-linear control

B. *Ant algirithm*

Route optimization is one method copied from ant behaviour. Normally, when ants are carrying food, they deposit a chemical called pheromone. Random routes between the nest and food are marked with pheromone. The shortest and fastest route soon has more pheromone than the other routes. Therefore, the ants can choose the most frequently used route shown in Figure 4. Pheromone has a limited lifetime, so only the shortest route has pheromone and it is soon the only way used between the nest and the food source [3].

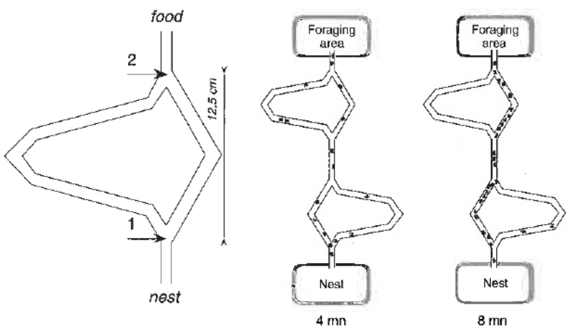


Figure 4: Practical ant algorithm

The digital solution of the ant algorithm could be usable in proactive routing, in which the agent messages travel via a random route between the source and the destination, leaving "digital pheromone" in the nodes. Then the most marked route is the shortest route, which the message frames with the payload can use.

C. *Neural network*

The idea for the neural network comes from the nervous system of animals. The activities of a neural network divide into many similar neurons. A single neuron reads signals from selected neurons and calculates its own value with simple rules for signaling to other neurons. The first layer of neurons uses input signals measured from the ambience and outputting results to the next layer. The next hidden layer uses the output of the previous layer as input and calculates values forward to the next layer. The output of the last response layer is the output of the complete neural network. One example is shown in Figure 5. The simplest neural networks include only one response layer [4].

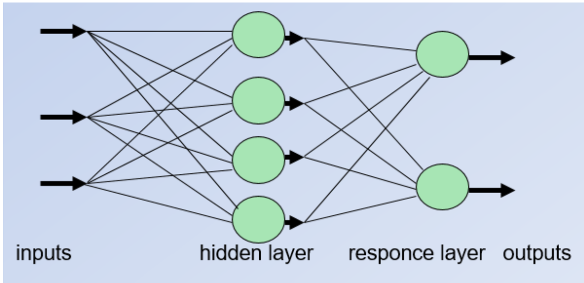


Figure 5: Neural network structure

The most interesting feature in a neural network is self-adapting: The learning process uses a large amount of learning material as example input-output pairs measured from the real system. The comparison between the learning material and input-output values of the neural network modifies the connection weights between neurons. After executing this comparison in numerous iteration loops, the neural network finally generates real output signals from the inputs, based on the learning material (Figure 6). This way the neural network creates a model of the real system. This is the most usable neural network usage: to create models of physical systems [4].

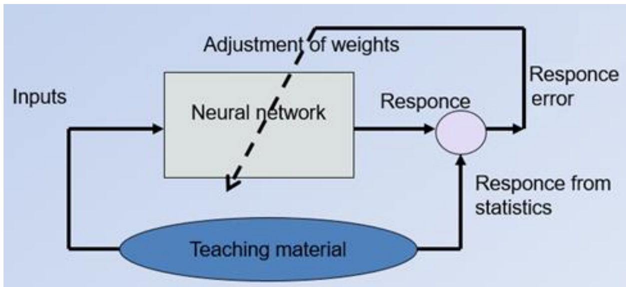


Figure 6: The learning process of Neural network

One interesting usage of the neural network is the prediction application to predict some value varying with time. In this case, the input of teaching material is historic values and the output the current value. The input of the working neural network is historic values including the current value. The output is the predicted value as seen in Figure 7.

In the prediction application, the learning process from history and predicting can continue parallel, at the same time. Therefore, over time, it learns to predict still better.

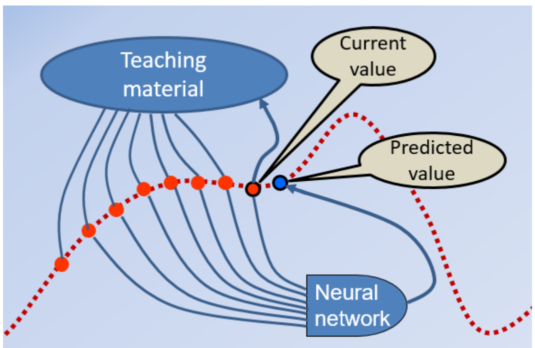


Figure 7: Prediction application

III. EXCAVATOR CALIBRATION SIMULATION

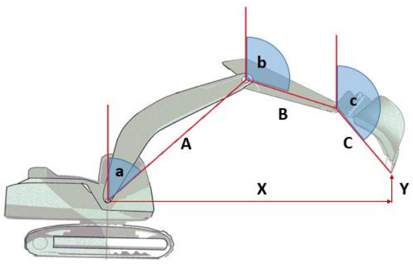
The exact use of the excavator requires some kind of depth measuring system to know the position of the bucket tip and to ensure a good result with the excavating work. The excavator needs a depth measuring system, for example when preparing the base of building foundations or digging a diversion ditch. The accuracy of the results depends on the measured dimensions of the excavator bars and the accuracy of the tilt sensors for every bar.

Wireless sensors simplify the installation, but anyway, there are a lot of installation stages, which requires accurate manual work. The installation of the measuring system requires care when assembling tilt sensors and measuring bar lengths. The different excavating work phases use different bucket forms and lengths. The tilt sensor is located normally in the bucket connector, so the buckets have different angles. This all changes the depth calculating parameters. While moving the measuring system to the other excavator, all the installation phases must be done again. The accurate setup of the installation takes a lot of time and gives possibilities for many errors.

The idea of the automatic tuning method is to let the excavator measure and tune itself. It already has sensors and with neural computing this kind of automatic tuning method is possible.

A. *The depth measuring method*

The measuring system requires the length and the tilt of every bar: the main bar, the arm bar and the bucket. The most used tilt sensor measures the acceleration between the sensor direction and the direction of gravity. Therefore, the sensor gives the sine or cosine of the angle in relation to the direction of gravity. The horizontal or the vertical dimension of the bar is the length of the bar multiplied by the sine or cosine of this angle. The principle of bucket tip position calculation is in Figure 8.



$X = A \sin(a) + B \sin(b) + C \sin(c)$ $\quad$ $Y = A \cos(a) + B \cos(b) + C \cos(c)$

Figure 8: Excavator deep measuring method

In a normal case the main bar length (A) and the arm bar length (B) are measured and are constant. The bucket length and the angle vary during the work stage. All angles can vary at installation. Because when installing a measuring system in a new excavator, all these values are variable.

B. *Calibrating with neural network*

While executing automatic tuning with a neural network, it needs learning data as an example of the right measured data, as mentioned earlier. In this case, the learning data consist of some fixed, predefined locations pointed to by the bucket tip in the learning phase. The learning phase tunes the weights or features of a neural network to fit with the learning material in the iteration loop. The method controls the activity of weight adjustments using coefficients. Later in real measurement cases, the system uses the found weights when calculating the real position data.

In this excavator case, the weights are the unknown data of the depth calculation: bar sensor angles and the bar lengths. The teaching material is a set of fixed reference test points. By the teaching iteration loop, the optimal method finds the actual sensor angles and bar lengths without manual tuning and measuring work.

In a practical teaching phase, the excavator driver points to every test point with the bucket tip as accurately, as possible. The first trunnion of the main bar in the excavator body should be the origin. The accurate positioning of the excavator itself is not always possible, so the position of the trunnion is also unknown. Now the neural network has a max of eight weights or unknown features to find in the learning iteration loop:

1. Main bar angle error
2. Arm bar angle error
3. Bucket angle error
4. Main bar length
5. Arm bar length
6. Bucket length
7. Origin X-error
8. Origin Y-error

The block diagram of the learning phase of the tuning method using a neural network is in Figure 9. Test reference point locations are the fixed predefined coordinates of points, which the excavator driver points to with the bucket tip. Sensor values are the real measured sensor values from every reference point scaled to be sines or cosines of the bar angles. Sensor data includes angle errors and origin position error.

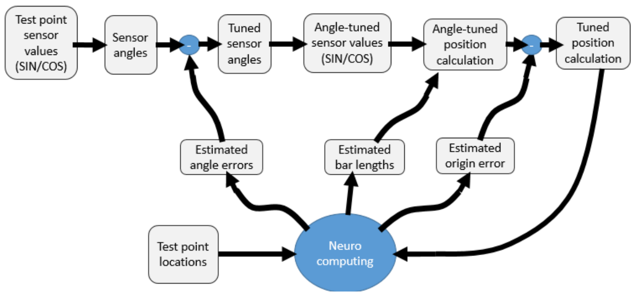


Figure 9: Excavator calibration with neural network

Calculation removes the estimated angle errors (calculated in the neural network) from calculated angles and again generates sine/cosine values of angles. Using the estimated bar length values, the tuning program calculates the estimated tuned bucket tip position. It removes the estimated origin error from the result to give the estimated test point coordinates. The neural teaching phase iteration compares these coordinates with the predefined fixed-point coordinates. This iteration gives more and more accurate values for the estimated data and hopefully finally finds just the right weights: angle errors, position error and lengths.

C. Calibration simulation

Simulation uses the tuning method with virtual data. As an input, simulation defines the initial reference test points including the origin error and the initial bar angles for every point. Using input data, the simulation generates initial bar angles for every bar at every test point and the corresponding initial sensor values, which are sine and cosine of the angles. The simulation adds the real angle errors to bar angles to generate simulated sensor data and corresponding accelerations. After that, the simulation calculates the default positions for the test points from the default sensor values. The points include origin error, which the program removes using default origin error to get the actual default test point coordinates. Then the simulation follows the teaching block flow described above. The block diagram of the simulation is in Figure 10. When the simulation is working property, after simulation the real angle errors, bar lengths and origin error are just the same as the corresponding estimated values.

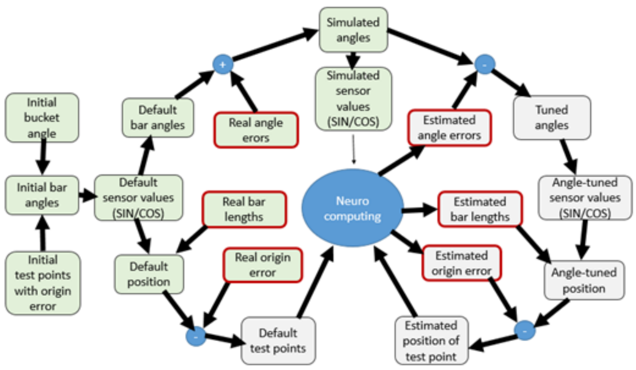


Figure 10: Calibration simulation

The simulations use three, four or five test points. While testing the simulation, angle errors over 45 degrees do not give the right results.

D. Simulation results

All simulations continue until the difference between the real and the estimated value is under 0.1 or as in the first 3-point test, the values are stable. The simulation includes three cases: in the first case, only the angles and origin error are unknown. In the second case, also the length of the bucket is unknown. In the third case, the lengths of all the other bars are also unknown. In every case, the reference test point coordinates are fixed and known. One case includes three parts with different coefficients and with 3, 4 or 5 test points found by trial runs. The output of the neural network computing is the condition: does it find the right features and how many iteration loops it needs.

In the most demanding case, where all bar lengths are unknown, the neural network learning phase gives the wrong results with 3 reference test points. While using 5 test points, it found, in addition to angels and the right origin, the lengths of all unknown bar lengths in 5688 iteration loops in Figure 11.

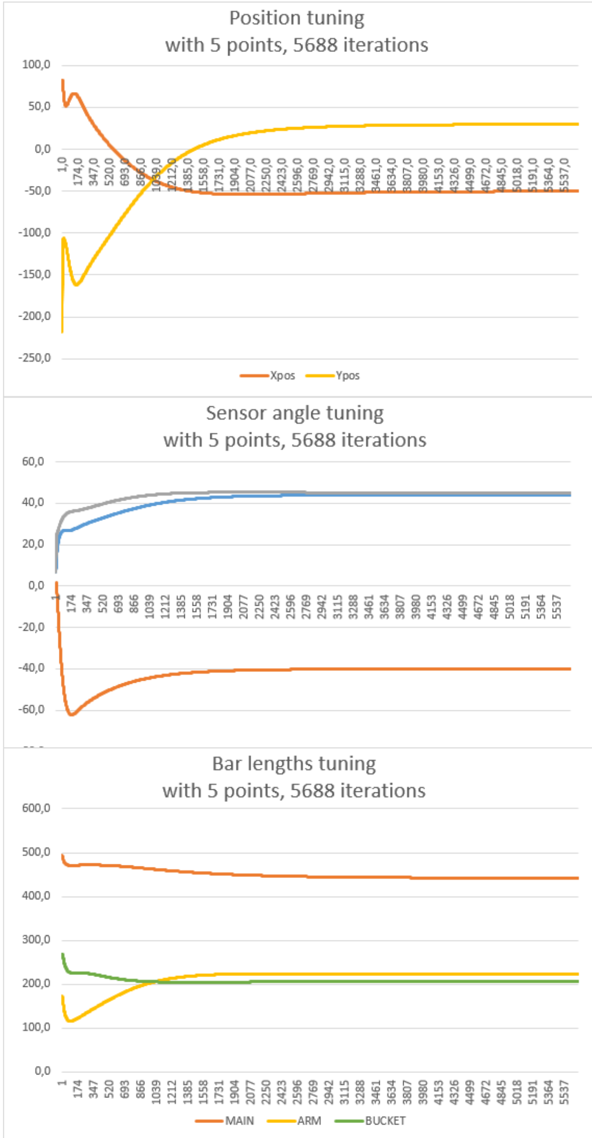


Figure 11: Calibration result example

All the results of three different cases, with 3, 4 and 5 test points, are in Table 1. As seen in the results, it takes a lot of calculation power also to find the lengths of all the bars in case 3. Therefore, the tuning method in case 3 is suitable for the first installation of the depth measuring system of the excavator. It perhaps requires a more powerful computing device, than a simple depth display. While the depth measuring system is ready to use, also a simpler processor can calculate features with 4 or 5 test points as in case 2 if the driver, for example, replaces the wireless acceleration sensors and changes to a different bucket for a new task.

Table 1: Simulation results

		Case 1	Case 2	Case 3
Fixed features (F) and Unknown features (U)	Test points	F	F	F
	Main bar	F	F	U
	Arm Bar	F	F	U
	Bucket	F	U	U
	Angles	U	U	U
	Origin	U	U	U
3 test points	Iterations	**3599**	**25673**	No results
4 test points	Iterations	**179**	**399**	**11878**
5 test points	Iterations	**117**	**86**	**5688**

The neural network found these results using selected default values for bar lengths, angle errors and origin error in Table 3. While tested with some other default real data, case 3 does not get any results. It seems that the excavator tuning method requires much more research work to function in a practical universal excavator application.

IV. Calibration Study Case

Automatic excavator tuning was tested with a simulation mentioned earlier. A student group at Seinäjoki University of Applied Sciences tested a part of the simulation in practice. The excavator for the test case was the small light model in Figure 12. The python-language program realized the neural network learning in this case. Three wireless acceleration sensors measured the sine and cosine of the angle of every bar. A USB-bridge functions as a virtual COM-port and collects the measured data from the sensors in Figure 13 [5].

Figure 12: The excavator model

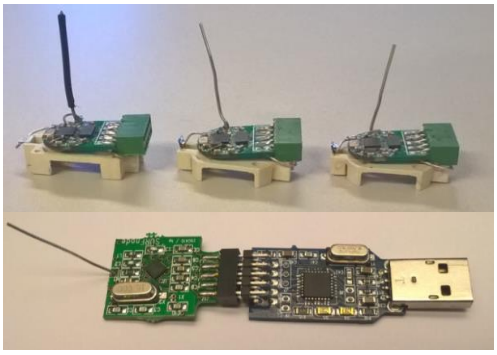


Figure 13: Acceleration sensors and USB bridge

The setup situation of the test case was that the lengths of every bar were known and measured. The sensor angles and the origin coordinates were unknown and should be defined by the neural network program. The practical test used 3 or 4 reference points for the bucket tip. Three different test cases showed how the practical tuning functions. In the first test, there were four reference points and angle errors: 180º, 103º and -140º and the origin error X=0, Y=6. The neural network did not find the result. The origin position and the Bucket angel did not stabilize.

In the second practical test case, the angle errors were limited to max ±30º: -5º, 15º and -8º, while the origin errors were X=3.2, Y=1.9. In this case, the neural network found the right results in 573 iteration loops (Figure 14).

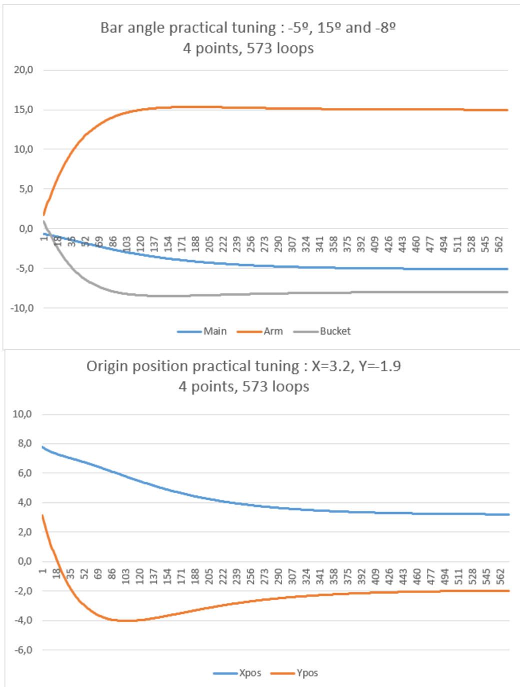


Figure 14: 4-point practical calibration results

The third practical test case includes only 3 reference points, but the angle errors were a max. ±30º: 13º, 15º and 29º. The origin error was: X=8.5, Y=20. Also, in this case the neural network found the right results, but needed more iteration loops: 763 (Figure 15).

The practical test cases follow the features of the simulation with maximum angle error limits of ±45º.

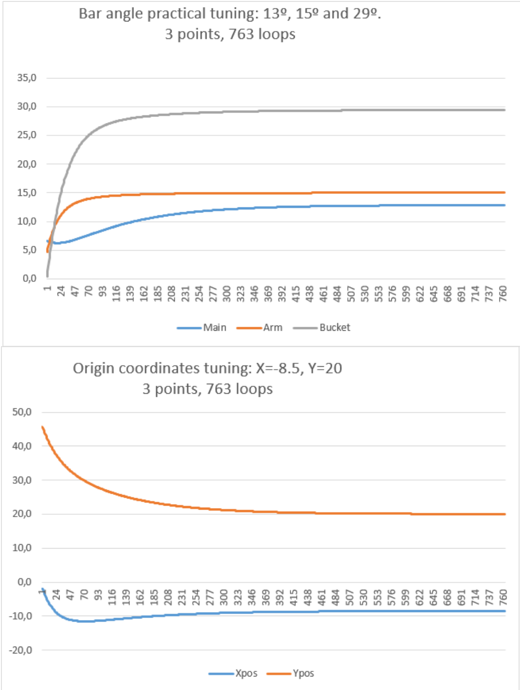


Figure 15: 3-point practical calibration results

V. CONCLUSION

Soft computing gives new possibilities for intelligent electronics. While the computing power of the low power micro controllers increase, the sensors can include a set of soft computing methods such as fuzzy logic, artificial intelligence or neural network. This article had shown that the simulations of the neural network were promising. The other result was the practical study cases, in which the new platform and simulation results were tested. The cases show that the development, research and simulations of soft computing were not only suitable in theory, but also in practice.

VI. REFERENCES

[1] H. Palomäki, Wireless Distributed Intelligence in Personal Applications, doctoral thesis, Vaasa: Vaasa University, 2017.

[2] V. A. Niskanen, Sumea logiikka, kirkasta älyä ja mallinnusta, Vantaa: Werner Söderström Oy, 2003.

[3] E. Bonabeau, M. Dorigo and G. Theraulaz, Swarm Intelligence: From Natural to Artificial Systems, New York, USA: Oxford University Press, 1999.

[4] D. Graupe, Principles of Artificial Neural Networks 2nd Edition, New Jersey: Word Scientific Publishing Co, 2007.

[5] H. Palomäki, "GENSEN project, New Platforms and Applications in Wireless Automation," in *10th International Symposium on Ambient Intelligence and Embedded Systems*, Chania, Crete, Greece, 2011a.