Neural Network based Control Method Implemented on Ambidextrous Robot Hand

Mashood Mukhtar1,*, Emre Akyürek1, Tatiana Kalganova1 and Nicolas Lesne2

1Department of Electronic and Computer Engineering, Brunel University, Kingston Lane, Uxbridge, London, UK
2Department of System Engineering, ESIEE Paris, Noisy-le-Grand Cedex, France

(Received 11 May 2016; Accepted 20 September 2016; Published on line 1 March 2017)

*Corresponding author: Mashood.Mukhtar@brunel.ac.uk
DOI: 10.5875/ausmt.v7i1.1171

Abstract: Human hands can precisely perform a wide range of tasks. This paper investigates key performance differences when conventional robotic hand controllers are combined with Neural Networks (NN). Tests are performed on a novel 3D printed multi-finger ambidextrous robot hand. The ambidextrous hand is actuated using pneumatic artificial muscles (PAMs) and can bend its fingers both left and right, offering full ambidextrous functionality. Force sensors are placed on the fingertips. In our control method, the grasping trajectory of each finger combines its data with that of the neighboring fingers to obtain accurate results.

Keywords: robot hand, ambidextrous hand design, grasping algorithms, control methods; pneumatic systems, multifinger control, neural network control

Introduction

Robotic manipulators have become increasingly important in the field of flexible automation. Neural Networks (NNs) can flexibly map nonlinear functions. Networks can be trained and applied both on or off-line. Of the many neural network types, two of the most widely used are multi-layer perception (MLP) and radial basis function (RBF) [1]. Back propagation is the most popular method of implementing multi-layer perception (MLP). There are three major learning paradigms: supervised learning, unsupervised learning and reinforcement learning. Each learning paradigm is suitable to solving a specific set of problems. A three-layer NN with full interconnections is shown in Figure 1.

The output of the two-layer NN is as follows:

\[ y_i = \sigma \left( \sum_{j=1}^{L} w_{ij} \sigma \left( \sum_{j=1}^{m} v_{ij} x_j + v_{i0} \right) + w_{i0} \right) \]  

where i = 1, 2, 3, ..., m, L is the number of neurons, \( \sigma \) is the activation function, and \( w \) is weight.

Neural networks have been widely applied in robot control and motion planning [2] [3]. They have been used to achieve motion control of manipulators [4], to help robots follow predetermined trajectories on city streets [5] and to achieve visual control of robotic arms [6]. A real-time learning neural robot controller was used to solve the inverse kinematics problem [7], and an artificial neural network was used to help a robotic arm system with six degrees of freedom to track and grasp a moving object [8].

Neural Networks can be implemented into robotic structures in several ways and with different controllers to provide improved solutions. For instance in [9], a learning process is designed for the two-links PAM manipulator to have an adaptive and dynamic self-organizing structure using NN and fuzzy logic. An NN was connected to PID loops in [10] to create an intelligent phasing plane switch control (PPSC) to overcome nonlinearities in PAM pressure feedback. NNs have also been integrated into particle swarm optimization to increase system accuracy [11].

The present paper combines neural networks with the PID, Bang-bang and Back-stepping algorithms. These controllers (PID, BSC, Bang-bang and SMC) are discussed in great detail in [12]. Table 5 summarizes the controllers combined with NNs and their respective experimental results. Force sensors are implemented on the fingertips.
of the ambidextrous robot hand. Use of these sensors with intelligent controllers increased robot hand autonomy as the grasping trajectory of each finger is based not only on its own feedback data, but also on that of the closest fingers.

\[ \text{Equation 1:} \]

The angles \( \theta \) and \( \theta' \) of the PAMs - the motors controlling the fingers - are experimentally defined to react to variations of the angle \( \theta \) of the object. If \( \theta \) is close to \( \pi/2 \), the fingers close to the object. If \( \theta \) is close to \( \pi/2 \), the fingers close to the object.

In Eq. 1, \( F_{cf} \) refers to adjacent fingers to both sides of concerned finger.

In Eq. 2, \( \theta_{f}(t) \) refers to the angle of each finger and a constant of 0.8 is the ratio experimentally defined to determine whether there is an abnormal increase of grasping angles.

\[ \theta_f(t) < 0.8 \times \theta_f(t) \] (3)

and

\[ \theta_{f+1}(t) < 0.8 \times \theta_f(t) \] (4)

If both (3) and (4) are correct, then the angle of finger \( f \) is much smaller than those of its adjacent fingers \( f - 1 \) and \( f + 1 \). Consequently, the more the finger \( f \) closes, the bigger \( \theta_f(t) \) becomes. Thus, a constant below 1 is used to check if \( \theta_{f-1}(t) \) or \( \theta_{f+1}(t) \) have stopped increasing at a smaller angle. If the finger \( f \) does not touch any objects, then it reverts to its vertical position. \( \theta_f(t) \) is then compared to a value close to \( \pi/2 \) to determine whether finger \( f \) is perpendicular to the palm. In so, finger \( f \) reverts to its vertical position without contacting any objects. These algorithms are summarized in Fig. 2. \( F_0 \) is the thumb, for which the angle is not considered.

\[ \text{Equation 2:} \]

The force feedback of each finger \( F_f(t) \) is the target force and \( F_f(t) \) is the force received from each finger. For the force feedback of each finger \( F_f(t) \), the values of the closest fingers \( F_{f-1}(t) \) and \( F_{f+1}(t) \) are also considered. In case \( F_f(t) = 0 \) but \( F_{f-1}(t) \) or \( F_{f+1}(t) \) receives a high force feedback, two different outcomes are possible. Either the object is not in contact with the sensor \( F_f \) or the object is not in contact with the finger at all. In the first case, the grasping controller must stop as the finger is actually in contact with the object. In the second case, not all fingers are needed to grasp the object. The detection of this case is translated as follows (where a constant 0.9 is the ratio experimentally defined to react to the object’s presence):

\[ F_{cf} \geq 0.9 \times F_t \] (2)

where \( F_{cf} = [F_{f-1}(t) \cup F_{f+1}(t)] \) and \( F_f(t) = 0 \)

If Eq. (2) is true, then at least one of the fingers close to the finger \( f \) is close to the object. If Eq. (2) is true and \( F_f(t) = 0 \), then the object is either not in contact with the sensor or with the finger. So the grasping controller must either stop or make the finger return to its vertical position. In the finger \( f \) is in contact with the object, the reacts differently by reading the angular angle. Thus angular feedback is read in reference to the angle of the vertical position. In Eq. 2 \( F_{cf} \) refers to adjacent fingers to both sides of concerned finger.

Results obtained with NNs

Table 1-4 shows the result obtained with the proposed approach, while Figs. 3 and 4 show the hand grasping a ball and a water bottle. The position of the fingers changes depending on the shape of the object being grasped. Table 1-4 also summarizes the angles and force received by the different fingers during grasping. The force target is 2.25 N ± 10% for the bottle and 1 N ± 5% for the ball. The indicated angles are those of the proximal phalanges.
When the hand grasps the bottle, the fingers come into contact with the object within 0.2 sec, but the angles continue to increase until 0.45 sec because the fingers continue closing until the bottle is pressed up against the thumb on the far side.

The middle finger is longer than the others, thus the force sensor on the tip of the middle finger does not come into contact with the object. However, because of the implemented NN, the force data collected from the neighboring fingers also play key part in the grabbing process as shown by the angle reached in Table 2.

The fingers react in a totally different way when grabbing a ball. The forefinger comes into contact with the object at 0.1 sec and its movement stops at 0.3 sec (as opposed to 0.4 sec for the bottle), as the object is bigger and the target force is smaller. Also, this grabbing action only involves the thumb and forefinger. As seen in Figs. 5 and 6, the different finger shape results in the middle finger having the slowest movement, whereas the little finger is the fastest.

As it applies no force and its angle becomes much smaller than that of the forefinger, it is deduced the middle finger is not in contact with the object. Therefore the finger starts rising before 0.3 sec. Next the NN is applied in the same way to the ring finger at 0.4 sec, and finally the little finger starts returning to its vertical position at 0.5 sec. The little finger moves more slowly than the middle and ring fingers because compressed air is already being used to drive their movement. The speed of the middle and ring fingers barely varies, as the PAM is in the middle of their contraction. Thus, a small increase of pressure still implies an important variation of the PAMs' lengths. The movement speed of the little finger increases at 0.8 sec, when the middle finger approaches the vertical position and has its own speed reduced. The compressed air is therefore only involved in the movements of the ring and little fingers. Finally, only the forefinger maintains its closing position, whereas the middle, ring and little finger return to their vertical positions. While the grabbing movement for the bottle was completed in 0.45 sec, that for the ball took 1 sec because it comprised both closing and opening movements.

### Experimental Analysis

Table 5 compares different behaviors observed with other grasping algorithms (SMC, PID, Bang-bang and BSC) developed at earlier stages. The sliding-mode control (SMC) introduced in [13] uses a grasping loop combined with average rising times, percentages of overshoot, number of oscillations, grasping times and settling times. SMC and other methods differ in significant ways. For instance, SMC runs in parallel, thus the percentage of overshoot is not applicable. The grasping and settling times are the same for the SMC because no backward control is implemented as in bang-bang control. However, the grasping and settling times are different for the bang-bang control because of the algorithm’s low sensitivity. Generally, SMC is easier to implement from a mechanical point of view, whereas PID is easier to implement from an algorithmic point of view. The implementation and calibration of grasping algorithms receiving feedback from the force sensors is much faster than from pressure transducers as the hysteresis of PAMs does not need to be taken into account with force sensors.
Bang-bang control is the fastest algorithm but also the least efficient one. It is not smooth enough to adapt itself to object shapes and can crush them. As explained in [15], the shooting function of the bang-bang controller is usually regularized with additional controllers. However, bang-bang control can be used to grab heavy objects. The higher the PAM pressure, the slower the PAMs contract, which is why their elasticity automatically opposes the shooting function.

BSC may be the most accurate algorithm, but is also the slowest one. As for PID control, BSC permits the fingers to adapt to the shape of objects with backward movements. Nevertheless, through the use of proportional and integrative controls, PID loops allow the fingers to move faster. The combination of PID control and SMC results in the accelerated rising time with SMC. As for conventional SMC, BSC depends on derivative and double derivative controls. This is the reason why the grasping time is much higher with the BSC, as it is not combined with proportional or integrative controls. Therefore, it takes 0.39 sec for the fingers to stabilise themselves with BSC, against 0.23 sec for SMC, 0.25 sec for PID control and

---

Table 1. Force (N) against time (sec) at the fingertips when the hand grabs a bottle.

<table>
<thead>
<tr>
<th>Time (sec)</th>
<th>0</th>
<th>0.05</th>
<th>0.1</th>
<th>0.15</th>
<th>0.2</th>
<th>0.25</th>
<th>0.3</th>
<th>0.35</th>
<th>0.4</th>
<th>0.45</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forefinger</td>
<td>88</td>
<td>79</td>
<td>65</td>
<td>39</td>
<td>30</td>
<td>25</td>
<td>18</td>
<td>13</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>Middle finger</td>
<td>93</td>
<td>84</td>
<td>75</td>
<td>44</td>
<td>32</td>
<td>27</td>
<td>20</td>
<td>11</td>
<td>10</td>
<td>8</td>
</tr>
<tr>
<td>Ring finger</td>
<td>86</td>
<td>76</td>
<td>67</td>
<td>36</td>
<td>29</td>
<td>23</td>
<td>16</td>
<td>11</td>
<td>10</td>
<td>9</td>
</tr>
<tr>
<td>Little finger</td>
<td>85</td>
<td>69</td>
<td>57</td>
<td>34</td>
<td>28</td>
<td>20</td>
<td>13</td>
<td>9</td>
<td>8</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 2. Finger angles (deg) against time (sec) when the hand grabs a bottle.

<table>
<thead>
<tr>
<th>Time (sec)</th>
<th>0</th>
<th>0.05</th>
<th>0.1</th>
<th>0.15</th>
<th>0.2</th>
<th>0.25</th>
<th>0.3</th>
<th>0.35</th>
<th>0.4</th>
<th>0.45</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forefinger</td>
<td>0</td>
<td>0.3</td>
<td>0.8</td>
<td>0.99</td>
<td>1.05</td>
<td>1.06</td>
<td>1.06</td>
<td>1.04</td>
<td>1.05</td>
<td>1.03</td>
</tr>
<tr>
<td>Middle finger</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Ring finger</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Little finger</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

---

Figure 5. Finger forces and angles against time when the hand grabs a ball.
0.20 sec for bang-bang control. Indeed, as for SMC, the main advantage of BSC is its ability to regulate nonlinear actuators. This is the reason why these two algorithms receive feedback from pressure or position sensors, as in [16]. Nevertheless, in our case, the feedback is received from force sensors directly implemented on the mechanical structure instead of the actuators themselves, as in previous research [17][18].

Conclusion

This paper presents a feasibility analysis for combining conventional controllers with neural networks. While conventional methods such as PID control are widely used and have been found to be reliable, combining them with artificial intelligence approaches offers better accuracy rates. All the tests are carried out on a novel 3D printed multi-finger ambidextrous robot hand. Force sensors are used to trigger the algorithm. The grasping trajectory of each finger is combined with data with the adjacent fingers to improve accuracy. Tables 1 to 4 and Figs. 5 and 6 show the finger force against time and angle. Table 5 presents testing results. Neural Networks are found to be useful in control applications and could be used as a safeguard against conventional controller failure.

Acknowledgment

The authors would like to cordially thank Anthony Huynh, Luke Steele, Michal Simko, Luke Kavanagh and Alisdair Nimmo for their contributions in the project.

Table 5. Performance comparison of conventional controllers when combined with NN.

<table>
<thead>
<tr>
<th>Grasping algorithms</th>
<th>Algorithms to which it is combined</th>
<th>Averaged rising time (sec)</th>
<th>Averaged % of overshoot</th>
<th>Averaged averaged grasping time (sec)</th>
<th>Averaged settling time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMC</td>
<td>PID, PPSC</td>
<td>0.20</td>
<td>N/A</td>
<td>0.23</td>
<td>0.23</td>
</tr>
<tr>
<td>PID</td>
<td>NN</td>
<td>0.16</td>
<td>5.3%</td>
<td>0.20</td>
<td>0.25</td>
</tr>
<tr>
<td>Bang-bang</td>
<td>Proportional, NN</td>
<td>0.10</td>
<td>40%</td>
<td>0.15</td>
<td>0.20</td>
</tr>
<tr>
<td>BSC</td>
<td>[14]</td>
<td>0.29</td>
<td>2.3%</td>
<td>0.37</td>
<td>0.39</td>
</tr>
</tbody>
</table>

Figure 6. Finger forces and angles against time when the hand grabs a ball.

References


[4] L. Tian, J. Wang, and Z. Mao, “Constrained Motion Control of Flexible Robot Manipulators Based on Recurrent Neural Networks,” IEEE Transactions on


