

Versatile In-Hand Manipulation of Objects with Different Sizes and Shapes Using Neural Networks

Satoshi FUNABASHI, Alexander SCHMITZ, Takashi SATO,
Sophon SOMLOR and Shigeki SUGANO, Fellow, IEEE

Abstract— Changing the grasping posture of objects within a robot hand is hard to achieve, especially if the objects are of various shape and size. In this paper we use a neural network to learn such manipulation with variously sized and shaped objects. The TWENDY-ONE hand possesses various properties that are effective for in-hand manipulation: a high number of actuated joints, passive degrees of freedom and soft skin, six-axis force/torque (F/T) sensors in each fingertip and distributed tactile sensors in the soft skin. The object size information is extracted from the initial grasping posture. The training data includes tactile and the object information. After training the neural network, the robot is able to manipulate objects of not only trained but also untrained size and shape. The results show the importance of size and tactile information. Importantly, the features extracted by a stacked autoencoder (trained with a larger dataset) could reduce the number of required training samples for supervised learning of in-hand manipulation.

I. INTRODUCTION

For achieving a manipulation task, adjustments to the initial grasping posture are often required. For example, after picking up a pen it is necessary to set the pen in the appropriate position in the hand before beginning to write. To achieve stability while changing the object's position during in-hand manipulation, the current tactile state has to be taken into account, especially if such movements include slip or soft materials. Generally, in case the hand or object have soft surfaces, it is difficult to find analytical solutions for changing the grasping posture. Moreover, it would be preferential if the desired behavior could be learned rather than programmed.

In previous research, the TWENDY-ONE hand, which has a special mechanical design inherently beneficial for in-hand manipulation, was used to achieve stable in-hand manipulation by simple interpolation control [1]. However, without tactile sensor feedback, i.e., the six-axis force/torque (F/T) and skin sensors, the in-hand manipulation was unstable. In particular, the final grasp depended on the initial grasp and if the initial grasping point was not correct the hand

This research was supported by the JSPS Grant-in-Aid for Scientific Research (S) No. 25220005, JSPS Grant-in-Aid for Young Scientists (B) No. 17K18183, the JSPS Research Fellowship for Young Scientists (DC) No. JP17J10571, the Research Institute for Science and Engineering of Waseda University and the Program for Leading Graduate Schools (Graduate Program for Embodiment Informatics) of MEXT.

Satoshi Funabashi, Alexander Schmitz, Sophon Somlor and Shigeki Sugano are with the Sugano Lab, School of Creative Science and Engineering, Waseda University, Tokyo, 169-8555, Japan (e-mail: s.funabashi@sugano.mech.waseda.ac.jp, schmitz@aoni.waseda.jp).

Takashi Sato was with the Sugano Lab, School of Creative Science and Engineering, Waseda University, Tokyo, 169-8555, Japan.

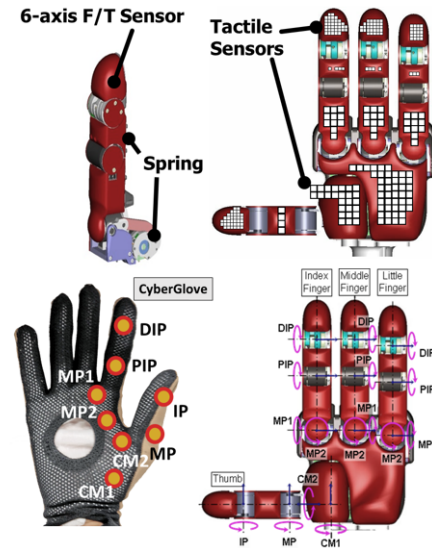


Fig. 1: (Top row) The hand of the human symbiotic robot TWENDY-ONE Hand. (Bottom row) The configuration of TWENDY-ONE Hand and CyberGlove.

dropped the object. Using a neural network (NN) for in-hand manipulation control, the area of initial grasping points for which the hand could achieve stable manipulation could be extended, but the learned network was specific to one size and shape of the object [2]. In general, even though a growing number of research on in-hand manipulation exists, stable in-hand manipulation for variously sized and shaped objects remains an open research problem.

In this paper we address the problem of versatile in-hand manipulation by deploying an improved NN design. Feed-forward NNs and deep NNs are used to enable the robot to manipulate objects of various sizes and shapes within its hand. As the robot hand grasps the object, the object's size can be extracted, which turned out to be an important feature to be used as input for the NNs. Hence, the desired movement is learned from a dataset including touch state and object size. This paper is an extended version of [3]. In [2][3] we already showed that the learned controller provides more robust in-hand manipulation than a pre-programmed, object-size specific position-controlled approach (i.e. interpolation control [1]). In the current paper, in addition to [3], cubes were manipulated to further demonstrate the robustness for non-round objects. Furthermore, feature extraction by stacked autoencoder is elaborated on to clarify the effective-

ness of using deep learning. Additional experiments compare the performance to using Principal Component Analysis (PCA). Moreover, in-hand manipulation with objects from the *YCB Object and Model Set* [4] are added.

II. RELATED RESEARCH

A. Previous Research on In-Hand Manipulation

Analytical solutions for in-hand manipulation of a sphere are provided in [5][6], but simplifying assumptions are made, such as rigid bodies, known geometries, no slip, point contacts and fingertips with six degrees of freedoms (DOF). In general, motion planning in complex environments with multiple constraints is a well-known problem [7][8]. Others have used a data glove to train in-hand manipulation and form compact grasp representations [9][10]. In [11] a dataglove and a genetic algorithm were used to learn in-hand manipulation. A Markov Decision Process for modeling and planning high-level in-hand manipulation has also been used [12]. A combination of modeling and machine learning is also investigated [13][14]. Moreover, enveloping grasps were investigated [15]. Others have achieved in-hand manipulation (in particular in-hand rolling and elevation) without sensors due to specialized robotic fingertips [16]. Often the current tactile sensor state of the robot is not taken into account. Yet, the importance of tactile sensing for object manipulation is well known [17]. Realistic contact modeling for object manipulation has been attempted [18][19], yet it is still challenging to achieve. Tactile information has been used for finger adjustment during in-hand object manipulation [20][21]. With the aim of more versatile handling, tactile sensors were used for a rolling contact of an unknown object [22]. A high-speed multi-fingered hand with a high-speed vision system has shown skills that exceed human capabilities for certain tasks [23]. Learning objects' impedance with the Allegro hand robustly achieved in-hand manipulation [24]. Reinforcement learning is also useful for manipulation [25]. However, each of those methods requires to learn a model for each object. In general, even though a growing number of research on in-hand manipulation has been performed, in-hand manipulation of variously sized and shaped objects remains an open research problem.

B. Deep Learning

The multimodal tactile information for in-hand manipulation used in this paper is high dimensional, which can be problematic for learning. Therefore, dimensionality compression mechanisms could be beneficial. However, it is difficult to manually integrate all the information by identifying and extracting the sensory features from each of the sensory modalities that are indispensable for robust in-hand manipulation. Some related research concentrated on PCA for classifying grasping postures [26]. We believe, however, that in-hand manipulation includes highly non-linear information that is difficult to identify via PCA.

Deep learning has recently attracted increasing attention also in the robotics community and will be used in this paper and compared to PCA. For example, Andrychowicz

et al. have recently achieved to learn dexterous in-hand manipulation, but used a setup with 19 cameras [27], while we focus on learning from tactile information. Considering the network architecture, we deployed a combination of Stacked AutoEncoder (SAE) and Feedforward NN to generate controlled in-hand manipulation. In particular, we used deep learning for the extraction of the features provided to the feedforward NN.

III. ROBOTIC SYSTEM

A. TWENDY-ONE Hand

The robot TWENDY-ONE [28] has hands which have 16 DOF each, as depicted in Fig. 1. The DIP and PIP joints of the index, middle and little finger are linked, and the hands are actuated by 13 small electric motors implemented in the joints of each finger. Each finger has a fingernail so that the hand can grasp, for example, a pencil on a desk. The DIP and MP1 joints also include springs, but there is no spring for the thumb. For the joints with springs, the actual joint angles can be calculated as the motor angles minus the spring displacements. Moreover, a soft skin made from silicone covers the whole surface of the hands. Therefore, the hands can compensate an error in the hand posture when grasping or manipulating an object. Moreover, the hands have many potentially useful sensors for in-hand manipulation. 241 distributed tactile skin sensors cover the whole surface of the hand. In addition, 6-axis F/T sensors are included in each fingertip. Each sensor has 1/256 (8 bit) resolution. The hand is about 20 cm long and the palm is 10 cm wide. In this research, only the thumb and the index fingertip were used, so only the distributed tactile sensors and 6-axis F/T sensors in those fingertips were used. Consequently, the 3 motors of the index finger and the 4 motors of the thumb were controlled.

B. Data Collection

There are several methods to get training data for the neural network. In our case, the fingers of the TWENDY-ONE hands are moved by tele-operation with a CyberGlove (22-sensor model from CyberGlove Systems) to record examples of successful in-hand manipulation. As the human tele-operates the robot hand, each motion has small differences, which creates variety for the training set. Amongst others, the dataglove has three flexion sensors per finger and four abduction sensors [29]. In order to map the sensor measurements to the thumb and index finger of TWENDY-ONE, the most distal index flexion measurement is ignored, and the proximal thumb flexion and thumb abduction sensor are added to move the CM2 joint of the robot. The proximal thumb flexion is also used for the robot's CM1 joint. For the other joints of the thumb and index finger there is a clear correspondence between a sensor measurement of the human hand and an actuated DOF of the robot. A left TWENDY-ONE hand and a right-hand CyberGlove was used so that it is easy for the experimenter to generate motions like in a mirror.

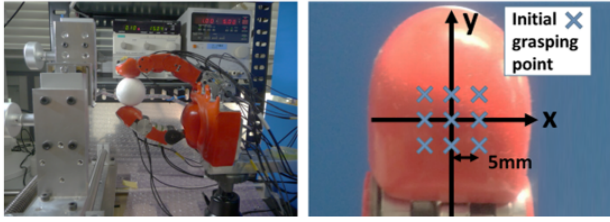


Fig. 2: Test setup with XYZ stage.

IV. METHOD AND SETTINGS

A. Experiment Design

The experiment setup is shown in Fig. 2. To control the positions on the fingertips where the object should be placed, an XYZ positioning stage was used. The position of objects can be determined in x, y and z axis in steps of 0.1mm. The starting position of each trial was chosen randomly by the experimenter, and it was attempted to gain training data with as many starting positions as possible so that neural network can learn effectively. The target movement in our experiments for various sized and shaped objects is shown in Fig. 3. The position of the objects should change from the bottom of the index finger to its side. This movement was chosen due to its high difficulty among many in-hand manipulation movements. In particular, it is difficult even during tele-operation to achieve the goal posture without dropping the object. In order to gather training data, the target movement was performed through tele-operation by dataglove with spheres and cylinders of diameter 20, 40, and 60mm. As shown in Fig. 4, 2 shapes and 3 diameters of objects were used for getting learning data. Each one was manipulated 50 times successfully. 300 success trials were recorded in total. In order that the neural network can learn easily, some preprocessing techniques were used. At first, each of the 300 trials was divided into 50 time steps, resulting in 15000 time steps in total. Time steps with a low resultant force were removed to avoid unstable grasps. Furthermore, random down-sampling was used. When recording the in-hand manipulation, there are times when the hand does not move, and this could negatively affect the network training. Therefore, Euclidian distance was used to cut off such redundant time steps as much as possible. After using those methods, 4734 time steps remained, and were used for training the neural network. The values of all sensor measurements were normalized to values between -1 and 1. Concerning the object parameters, the size information also was normalized (example values in Fig. 4). The object size was not provided, but when the hand grasps an object, the object size is calculated using a kinematic model of the hand and the joint angles. The estimated sizes were used for the size of objects the neural network learns. Furthermore, the shape information was coded as -0.8 for the sphere and +0.8 for the cylinder.

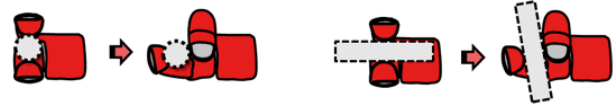


Fig. 3: The in-hand manipulation that should be performed, shown for a sphere and a cylinder. The successful manipulation is that the position of the two fingertips changes from being on a vertical line to a horizontal line.







Object						
Diameter mm	20	40	60	20	40	60
$p_1(O)$ (Size)	-0.8	0.0	0.8	-0.8	0.0	0.8
$p_2(O)$ (Shape)	-0.8			0.8		

Fig. 4: Object setting for learning dataset.

B. Neural Network

A feedforward neural network (FNN) was used. Normally, the inputs of the FNN for in-hand manipulation include joint angles and touch states and the network generates the next time step of joint angles. In this research, size and shape information of an object are added as inputs for the FNN as shown in Fig. 5. The hyperbolic tangent is used as an activation function. The stochastic gradient descent is used for the optimizer of the FNN with a learning rate of 0.0001, L2 loss function with L2 regularization ($\lambda = 0.0001$) and a minibatch size of 100. Initial momentum for the stochastic gradient descent is 0.5 and it is gradually changed. After 2000 epochs, the momentum becomes 0.9. To implement the network, the Theano library for python and GTX Geforce 580 as GPU were used, also for the deep learning. As explained in the next section, we also attempted to use the network (during training and testing) with a subset of the totally available inputs without size and shape information, with only size information and with only shape information. One hidden layer with 100 neurons was used. The stacked autoencoder (SAE) will be explained in the next section.

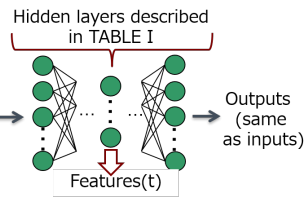
C. Stacked AutoEncoder

As depicted in Fig 5, the input for our Stacked AutoEncoder (SAE) has a dimension of 93, i.e., excludes information about the object's size and shape. The idea behind SAEs is to reproduce their own inputs in the output layer throughout the center layers in which the features are compactly represented by reducing the number of dimensions in the center layer. These acquired features are used for the FNN as input instead of the original input. The number of hidden layers and the number of nodes in each hidden layer of the SAE depend on the number of extracted features. The design of the hidden layers of the SAEs are specified in Table I.

Out of the 15000 time steps recorded, 10000 time steps were randomly chosen for unsupervised layerwise training. Although deep learning methods usually require large

SAE

Inputs(t) (93 inputs in total)
 7 joint angles
 2 spring displacements
 2*6 6-axis F/T sensors in the fingertips
 2*36 distributed sensors in the fingertip skin



FNN

Inputs(t) (93 inputs in total)
 or
 Features(t) (2~93 dimensions)

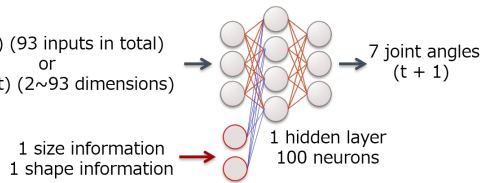


Fig. 5: Top: Stacked AutoEncoder (SAE) for feature extraction. Bottom: FNN for generating motion. The inputs and outputs for the two networks are shown. Depending on the setting, the FNN uses the features calculated by the SAE.

TABLE I: Stacked-Autoencoder Settings

Features	Structure of hidden layer
90	90
80	85-80-85
70	80-70-80
60	80-70-60-70-80
50	80-65-50-65-80
40	90-75-55-40-55-75-90
30	90-70-50-30-50-70-90
20	80-60-40-20-40-60-80
10	70-50-30-10-30-50-70
5	70-50-30-10-5-10-30-50-70
2	70-50-30-10-2-10-30-50-70

amounts of training data, the current paper shows that 10000 samples indeed can be enough for learning in robotic applications. The activation function is the same as for the FNN. Learning rate and learning decay rate for pre-training were set to 0.01 and 0.99999, respectively. The number of epochs was set to 5000. For fine-tuning purposes, learning rate and learning decay rate were set to 0.1 and 0.999999, with the number of epochs set to 20000. For the both pre-training and fine-tuning, the stochastic gradient descent is used for the optimizer of the SAE with a L2 loss function and L1 regularization ($\lambda = 0.00001$) and a minibatch size of 100. Initial momentum for the stochastic gradient descent is 0.5 and it is gradually changed. After 2000 epochs, the momentum becomes 0.9. We also attempted to use a stacked denoising auto-encoder (SdA), as previous experiments suggested this as being beneficial for the tactile sensing using the TWENDY-ONE Hand. However, using dropout in the training resulted in higher errors in the reconstructed motions as verified with test data. Therefore, SdAs were not used in our current experiments.

V. EVALUATIONS

A. Effect of size and shape

Initially, the importance of different input information was evaluated. First the usefulness of size information was confirmed, and results showed that the object size as determined from the proprioceptive sensor readings of the initial pose as the input for the neural net was effective for the performance of learning and less training epochs were necessary. In this evaluation, the initial grasping position of the object was always the same ($x=-10\text{mm}, y=-10\text{mm}$) and a sphere-shaped object with 30mm diameter as the untrained object was used for this evaluation. When providing the size information, 10000 training epochs were needed. With 10000 training epochs and size information all the trials were successful. On the other hand, when the size information was not provided, after 10000 and even 30000 training epochs, the target manipulation was not correctly done and sometimes the hand dropped the object. In particular, without object size, after 10000 training epochs, regardless of the initial grasping position, no successful manipulation was achieved. For 30000 training epochs only 1 out of 10 trials (with the controlled initial grasping position $x=-10\text{mm}, y=-10\text{mm}$) achieved a successful manipulation. In 9 out of 10 trials the sphere was dropped from the hand or the required final posture was not achieved, i.e. the position of the two fingertips was not on a horizontal line.

Next, we investigated the importance of shape information. In short, without providing the shape information to the neural network, reliable behavior could be produced nevertheless. The starting position was the same as before ($x=-10\text{mm}, y=-10\text{mm}$) and the same size of sphere was used. In-hand manipulation with size and shape and only size was compared to each other. For size and shape information, the number of training epochs was 10000. The target manipulation was not achieved and also the sphere was falling from the fingertips. When increasing the number of training epochs from 10000 to 15000, the successful in-hand manipulation was done in all the 10 trials. On the other hand, in-hand manipulation with only size information was also successful. From this viewpoint, shape information is not beneficial for in-hand manipulation. On the contrary, if size and shape information were provided at a same time, depending on the number of training epochs, the results for in-hand manipulation can be worse than when providing only size information. All the results are shown in Table II. An example of a successful in-hand manipulation is shown in Fig. 6.

Importantly, in-hand manipulation of objects that are not grasped at the fingertip's center is often not successful with preprogrammed, object-size specific position-control [3], as already discussed in the introduction.

B. Grasping force with tactile information

In the second evaluation, the tactile information, which can be acquired by the 6-axis F/T sensors and distributed tactile sensors in the fingertips, was also found to be important for

TABLE II: Achievement of the final posture with 30 mm

Provided Information	Training Epochs	Success Rate
Size, no shape	10000	10/10
No size, no shape	10000	0/10
No size, no shape	30000	1/10
Size, shape	10000	0/10
Size, shape	15000	10/10



Fig. 6: Example of an in-hand manipulation for a sphere of diameter 30mm. Even though the initial grasping posture is out of the center of the fingertips ($x=-10\text{mm}, y=-10\text{mm}$) the handling was successful.

in-hand manipulation. In this evaluation spheres of diameter 30 and 50mm were used; the initial grasping position was $x=0\text{mm}, y=-10\text{mm}$. Two settings for the input of the neural network were compared, the first with providing only the motor angles and the spring displacements, and the second with providing the motor angles, the spring displacements, 6-axis F/T sensors and distributed tactile sensors. In all cases the objects were not dropped from the hand. In the first setting (without tactile information), the final desired grasping posture was not achieved with all tested sizes and shapes, as shown in Fig. 7. With the tactile information, the desired final grasping posture was robustly achieved in all 4 cases. In particular, Fig. 7 shows some examples of the in-hand manipulation without tactile information on the left side and with tactile information on the right side. Without tactile information, the final grasping posture is not correctly generated (the two fingertips are not in a horizontal line). Importantly, the resultant force is higher when not using the tactile information, which makes sense, as no force feedback is provided in this case. On the other hand, with tactile information the final grasping posture is correctly achieved, and the force magnitude is mostly under 40N.

From these evaluations we could confirm that size and tactile information is important for in-hand manipulation, while the importance of shape information could not be proven. Therefore, for further experiments we used only size and tactile information as input for the neural network. Another reason not to use the object shape as information for the neural network is that it is hard to get the object shape information from the sensor readings of the hand, in particular with 2 fingers, while the object diameter can be determined from the initial grasping posture using only

kinematics.

C. Generalization to untrained objects

In the third evaluation, spheres of diameter 20, 40 and 60 mm and cylinders of diameter 20, 40 and 60mm were used for training and those of 30 and 50 mm were used for testing. If the size is below 20 mm, the sphere is so light that it can stick to the surface of the hand. If the diameter is over 60 mm, it is barely physically possible for the hand to generate grasping postures correctly. In results after learning in-hand manipulation with 20, 40 and 60 mm of spheres and cylinders, it could generate in-hand manipulation with spheres of diameter 30 and 50mm and cylinders of diameter 30 and 50mm. As long as the initial grasping posture was kept within certain limits, the final desired grasping postures were achieved in all our trials with all sizes and both shapes. We tested each size and shape combination at least 5 times.

Furthermore, we investigated whether the egg-shaped object as a novel shape can be manipulated or not after training in-hand manipulation with spherical and cylindrical objects. The egg-like shape objects have intermediate curvatures between that of spheres and cylinders. The diameters of the egg-shaped objects are 30, 40 and 50mm. Furthermore, three initial grasping postures, at the sides, top and bottom, and diagonally were tested as shown in Fig. 8. We tested each pose with each size at least once, and also in this case robust in-hand object manipulation was achieved in all our trials. In particular, we tested pose B and C ten times each as we deemed them to be particularly challenging. Despite of different grasping postures all our trials were successful.

Moreover, cube-shaped objects of diameter 30, 40 and 50mm were used. Three initial grasping postures, at the sides, diagonally in one axis or diagonally in two axes were tested as shown in Fig. 8. We tested each pose with each size three times, and also here successful manipulation was achieved in all our trials. For the cube shaped object, we assumed pose B and C to be particularly challenging, but nevertheless they were robustly manipulated. For comparison, the egg-shaped and cube-shaped objects in Pose B and C were difficult to grasp with our stiff gripper in the XYZ stage, which demonstrates the difficulty of grasping the objects in those postures. As a further consequence, the experimenter handed those objects to the robot hand, which may have resulted in slight variations in the starting position, but it was aimed to hand the egg and the cube in the center of the fingertip.

VI. EVALUATIONS FOR DEEP LEARNING

A. Feature extraction

In the previous section the raw sensor values were used as input for the feedforward neural network, and the total input had a dimension of 93. If the size and shape information are included, the total input has a dimension of 95. Nowadays, deep learning methods for robotics show good results in compressing data and extracting higher level information from raw data. Robust movements of robots with deep neural networks were generated in various research, however for in-hand manipulation, such methods are rarely used. A

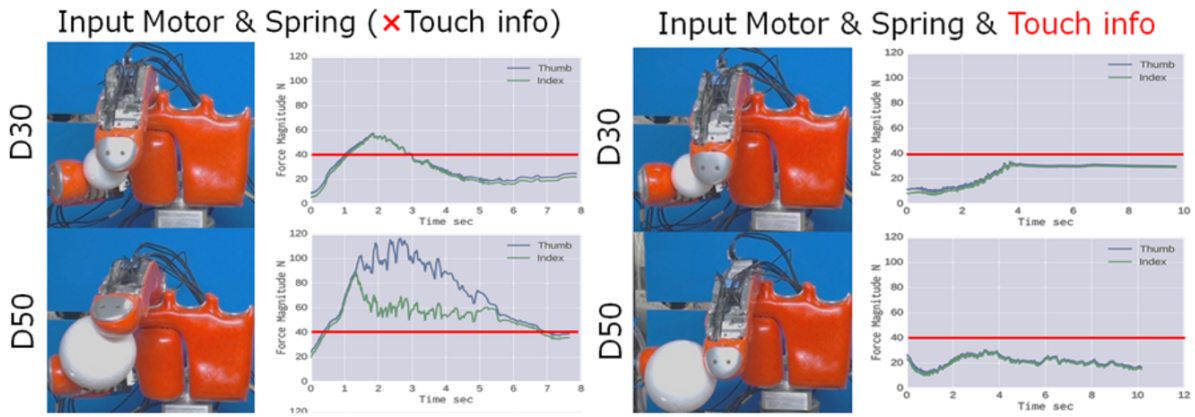


Fig. 7: In-hand manipulation after training with and without the tactile information. Without the tactile information (6-axis F/T sensors in the fingertips and distributed force sensors in the skin) the final desired grasping posture was not reached in any of our trials. The resultant force, as measured by the 6-axis F/T sensors in the fingertips, is shown (the range of the force in the graphs is from 0 to 120N). It can be clearly seen that the force without the information was higher. For reference reasons, the red line shows 40N, which was never exceeded when using the tactile information. The green line shows the resultant force in the index finger, the blue line in the thumb. In the cases where the resultant force is different in the thumb and index finger, probably overload occurred. The starting position in all these trials is the same ($x=0\text{mm}, y=-10\text{mm}$).

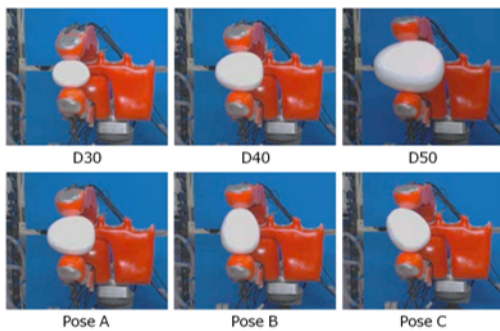


Fig. 8: An untrained, egg-like shaped object was also used to evaluate the robustness of the learned in-hand manipulation skill. This figure shows the different sizes and poses that were used for evaluation.

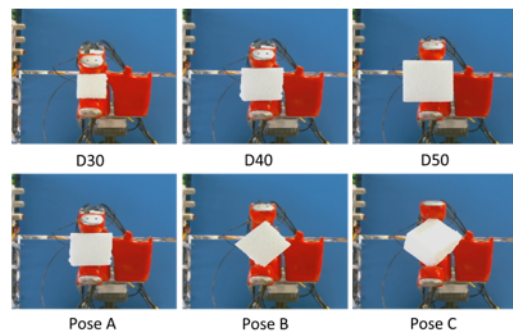


Fig. 9: An untrained, cube shaped object was also used to evaluate the robustness of the learned in-hand manipulation skill. This figure shows the different sizes and poses that were used for evaluation.

reduced input dimension with higher level features can show improved learning results, in particular less training samples could be necessary for the supervised learning.

Therefore, in this section we evaluated which number of features from the stacked auto-encoder (SAE) are useful, and when used as input for the FNN how many training samples are necessary with such reduced input dimensionality. Fig. 10 shows the mean squared error of the SAE with different number of units in the smallest hidden layer (the outputs of the hidden layer can be used as input features for the FNN). There are big changes around 20 and 60 minimal hidden layer size. When further investigating the result, with 20 neurons the error between regenerated and original joint trajectory is large. Therefore, 60 features is focused for controlling the actual hand. In the next section we will evaluate whether the extracted features from the stacked auto-encoder can produce a good performance when used as input for the FNN.

B. Real time control

The performance of a deep neural network for controlling the hand was evaluated. In this case, the activations of the smallest hidden layer of the SAE were used as input features for another neural network for supervised learning. According to the prior results in Fig. 10, we attempted to use the SAEs that compressed the raw input to 50 and 60 features, respectively. Accordingly, the supervised learning had either 50 or 60 inputs. The hidden layer size for the supervised learning was 100, as in the previous section. Several attempts with 50 input features for the supervised learning did not produce stable behavior, but with 60 features reliable object manipulation behavior could be achieved. We compared the success of the supervised learning when either using the raw sensor values (only normalized) or the features provided by the SAE. Only for the learning with the raw data the object size was provided in addition. 900 random training samples

were used for the supervised training. Even though the neural network with the raw sensor values provided very reliable results with 4734 training samples, it could not produce reliable behavior with only 900 training samples. The deep neural network on the other hand provided robust behavior. In particular, both the sphere and cylinder with diameter 20, 40 and 60 mm as presented in the last section were manipulated 5 times each, and all 30 trials were successful with the features from the deep neural network. In these experiments, the starting position of the objects was always in the center of the fingertips. The shallow neural network on the other hand never produced the wanted behavior. Fig. 12 shows examples of the typical behavior produced with the deep and the shallow neural network. With the deep neural network, a smooth motion was performed, which lasted for about 3 seconds, and then the hand stopped. However, with the shallow neural network the motion took much longer, and even after 60 seconds the hand did not stop moving, and the fingertips were swaying from side to side.

C. Generalization to untrained objects

The generalization capability to untrained objects is investigated. The objects are randomly placed on the fingertips. First, spheres and cylinders of 30 and 50 mm in diameter were manipulated successfully. Egg and cube shaped 30 and 50 mm objects are also manipulated. The network with 60 features from the SAE and 900 training samples could generate correct motions for untrained objects. From these results, the network seems to acquire the information of different sized and shaped of objects from 20 to 60 mm in diameter.

Furthermore, some real objects from daily life are used from the *YCB Object and Model Set* [4]. A golf ball with 42 mm, a rope with 40 mm, a plastic strawberry with 46 mm and a plastic cup with 54.5 mm diameter shown in Fig.11 were manipulated successfully in all trials (three times each). Even though they have a rough texture and complicated shape (especially the rope and the plastic cup have an asymmetric shape), the hand could robustly achieve in-hand manipulation. This suggests that the network can manipulate many real-world objects and does not necessarily need to consider the shape of the objects for the given task.

D. Comparison with PCA

Another commonly used feature extraction method, principal components analysis (PCA), was used to be compared to the SAE. A sphere and cylinder with a diameter of 40 mm are manipulated 5 times for each. The positions of those objects are randomly decided. The PCA extracted features with 60, 80 and 90 dimensions with 900 samples and 4734 samples for 60. As Table III shows, only the SAE could generate correct in-hand manipulation 10 times. For 60 dimensions of features extracted by PCA, the motion does not reach the final posture with 900 and 4734 samples and the hand sometimes drops the cylinder. For 80 and 90 dimensions of features extracted by PCA, with 80 features the sphere is dropped and it fails to reach the final posture with the

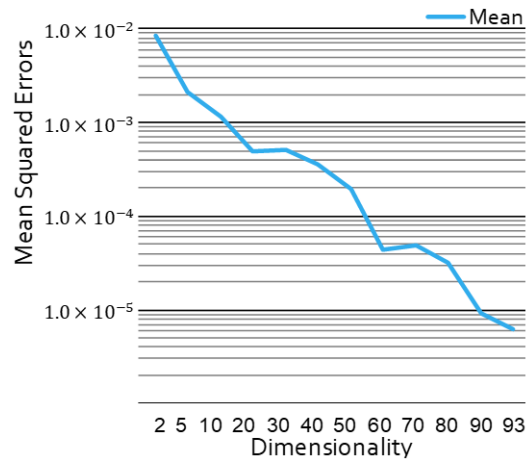


Fig. 10: Mean square errors for learning of the SAEs are generated. The SAE has 2 to 93 dimensions of hidden neurons. A large change occurs in 60 features as shown. It may be a sign that extracted information holds important in-hand manipulation with up to this number of hidden neurons.

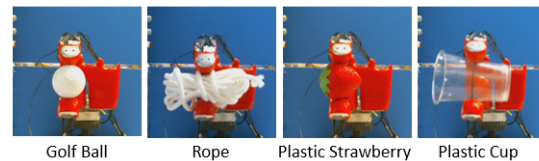


Fig. 11: Untrained real-world objects from the "YCB Object and Model Set" were correctly manipulated.

cylinder and even 90 features from the PCA could not generate correct motions but almost correct motions with a sphere, while dropping the cylinder. From these results, the SAE is more practical as a feature extraction method than the PCA.

VII. CONCLUSION

This paper presented a method for versatile in-hand manipulation of variously sized and shaped objects. The method enabled the robot hand to generate in-hand manipulation for objects of sizes and shapes that it had not been trained with. When using a deep neural network (trained with a larger dataset), the number of required training data for the supervised learning could be decreased. The deep network can extract more effective features than PCA. In-hand manipulation with real objects was also achieved. These results suggest daily tasks, for example re-grasping a pencil (even untrained one) before writing, can be achieved with our method. In summary, compared to prior results, we have extended the in-hand manipulation capabilities of the robot hand.

Within the work presented, we considered and trained only one specific in-hand manipulation skill. More versatile capabilities, i.e., a wider range of useful and differentiated in-hand manipulation skills would be desirable in the future. As the generation of training data and the learning itself can



Fig. 12: Examples of in-hand manipulation with 60 features (top 2 rows) and raw data (bottom 2 rows). The time duration for both motions is not the same: it took about 3 seconds with the 60 features to reach a static grasp, but with the raw sensor values the fingertips kept swaying from side to side.

TABLE III: Achievement of the final posture with 40 mm

Feature Extraction	Number of Inputs	Samples	Success Rate
SAE	60	900	10/10
PCA	60	900	0/10
PCA	60	4734	0/10
PCA	90	900	0/10
PCA	80	900	0/10

be burdensome, it is conceivable that transfer learning from one in-hand manipulation skill to another is advantageous. Some recent research also focuses on convolutional neural networks (CNN) and already established for robot vision, it might also be useful for tactile recognition. Therefore, using the CNNs for in-hand manipulation could be one further promising approach.

REFERENCES

- [1] H. Iwata, R. Hayashi, Y. Shiozawa, S. Sugano, "Basic Control Techniques of TWENDY-ONE Hand which has Passive Flexibility - Achievement of Diverse Grip and Manipulation by Transitions Between Grip Forms," 26th Conference of Robotics Society of Japan, paper no.1E3-05, Hyogo, September 2008 (in Japanese).
- [2] K. Kojima, T. Sato, A. Schmitz, H. Arie, H. Iwata and S. Sugano, "Sensor Prediction and Grasp Stability Evaluation for In-Hand Manipulation," 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2013.
- [3] S. Funabashi, et al., "Robust In-Hand Manipulation of Various Sized and Shaped Objects," 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.257-263, 2015.
- [4] <http://www.ycbbenchmarks.com/>
- [5] L. Han, Y. Guan, Q. Li, Z. Shi, and J. Trinkle, "Dextrous manipulation with rolling contacts," Proceedings of 1997 IEEE International Conference on Robotics and Automation, 1997, pp. 992-997.
- [6] L. Han and J. Trinkle, "Dextrous manipulation by rolling and finger gaiting," Proceedings of 1998 IEEE International Conference on Robotics and Automation, pp. 992-997, 1998.
- [7] S. M. LaValle and J. J. Kuffner, "Rapidly-exploring random trees: Progress and prospects," Algorithmic and Computational Robotics: New Directions, pp. 293-308, 2001.

- [8] B. Sundaralingam, T. Hermans "Relaxed-Rigidity Constraints: In-Grasp Manipulation using Purely Kinematic Trajectory Optimization", Robotics: Science and Systems (RSS), 2017.
- [9] R. Martins, D.R. Faria and J. Dias. "Symbolic level generalization of in-hand manipulation tasks from human demonstrations using tactile data information", 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems Workshop on Grasp Planning and Task Learning by Imitation, October, 2010.
- [10] G. Cheng, N. Hendrich, J. Zhang, "In-hand manipulation action gist extraction from a data-glove", Proceedings of IEEE International Conference on Cognitive Systems and Information Processing, 2012.
- [11] J. Gonzalez-Quijano, M. Abderrahim, C. Bensalah and A.Al-kaff. "A human-based genetic algorithm applied to the problem of learning in-hand manipulation tasks", 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems Workshop: Beyond Robot Grasping - Modern Approaches for Learning Dynamic Manipulation, October, 2012.
- [12] U. Prieur, V. Perdereau and A. Bernardino, "Modeling and Planning High-Level In-Hand Manipulation Actions from Human Knowledge and Active Learning from Demonstration", 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, October, 2012.
- [13] M. Liarokapis, and A. M. Dollar, "Learning Task-Specific Models for Dexterous, In-Hand Manipulation with Simple, Adaptive Robot Hands," 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2016.
- [14] M. Liarokapis, and A. M. Dollar, "Deriving Dexterous, In-Hand Manipulation Primitives for Adaptive Robot Hands," 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems 2017.
- [15] M. Kaneko, "Enveloping Grasp", The Robotics Society of Japan, Vol. 18, No. 6, pp. 782-785, September, 2000.
- [16] K.Tahara, et al.: "External sensorless dynamic object manipulation by a dual soft-fingered robotic hand with torsional fingertip motion", Proceedings of 2010 IEEE International Conference on Robotics and Automation, 2010.
- [17] R.D. Howe, "Tactile Sensing and Control of Robotic Manipulation", Journal of Advanced Robotics, 8(3):245-261, 1994.
- [18] A. Nakashima, T. Shibata, Y. Hayakawa, "Control of Grasp and Manipulation by Soft Fingers with 3-Dimensional Deformation", SICE Journal of Control, Measurement, and System Integration, VOL.2, NO.2, pp.78-87, 2009.
- [19] H. Scharfe, N. Hendrich, J. Zhang, "Hybrid physics simulation of multi-fingered hands for dexterous in-hand manipulation", Proceedings of 2012 IEEE International Conference on Robotics and Automation, 2012.
- [20] J.A. Corrales, F. Torres and V. Perdereau. "Finger Readjustment Algorithm for Object Manipulation based on Tactile Information", International Journal of Advanced Robotic Systems, Vol. 10. No. 9. pp.1-9. 2013.
- [21] R. Platt, A.H. Fagg, R. Grupen, R. "Null Space Grasp Control: Theory and Experiments", IEEE Transactions on Robotics, Vol 26, No 2, 2010.
- [22] H. Maekawa, K. Tanie, and K. Komoriya, "Tactile sensor based manipulation of an unknown object by a multifingered hand with rolling contact", Proceedings of 1995 IEEE International Conference on Robotics and Automation, May, 1995.
- [23] N. Furukawa, A. Namiki, S. Taku, and M. Ishikawa, "Dynamic regrasping using a high-speed multi fingered hand and a high-speed vision system", Proceedings of 2006 IEEE International Conference on Robotics and Automation 2006. Proceedings 2006 IEEE International Conference on, pp. 181-187, May, 2006.
- [24] M. Li, K. Tahara, A. Billard, "Learning Object-level Impedance Control for Robust Grasping and Dexterous Manipulation", Proceedings of 2014 IEEE International Conference on Robotics and Automation, 2014.
- [25] H. van Hoof, T. Hermans, G. Neumann and J. Peters, "Learning Robot In-Hand Manipulation with Tactile Features", 2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids) November 3-5, 2015.
- [26] Marco Santello, et al. "Postural Hand Synergies for Tool Use", The Journal of Neuroscience, 10105-10115, December, 1998.
- [27] M. Andrychowicz, et al. "Learning Dexterous In-Hand Manipulation", arXiv:1808.00177
- [28] H. Iwata and S. Sugano. "Design of Human Symbiotic Robot TWENDY-ONE", Proceedings of 2009 IEEE International Conference on Robotics and Automation, 2009.
- [29] <http://www.cyberglovesystems.com/products/cyberglove-ii/overview>