Available online at www.sciencedirect.com

ScienceDirect

Procedia Engineering 96 (2014) 10 – 20

Procedia Engineering

www.elsevier.com/locate/procedia

Modelling of Mechanical and Mechatronic Systems MMaMS 2014

# Advanced Robotic Grasping System Using Deep Learning

Pavol Bezak[a,*], Pavol Bozek[b], Yuri Nikitin[c]

[a]*Research Centre of Progressive Technologies, Slovak University of Technology, Hajdoczyho 1, 917 24 Trnava, Slovakia*
[b]*Institute of Applied Informatics, Automation and Mathematics, Slovak University of Technology, Hajdoczyho 1, 917 24 Trnava, Slovakia*
[c]*Department of Mechatronic Systems, Kalashnikov Izhevsk State Technical University, Izhevsk, Russia*

**Abstract**

Object grasping by robot hands is challenging due to the hand and object modeling uncertainties, unknown contact type and object stiffness properties. To overcome these challenges, the essential purpose is to achieve the mathematical model of the robot hand, model the object and the contact between the object and the hand. In this paper, an intelligent hand-object contact model is developed for a coupled system assuming that the object properties are known. The control is simulated in the Matlab Simulink/ SimMechanics, Neural Network Toolbox and Computer Vision System Toolbox.


Peer-review under responsibility of organizing committee of the Modelling of Mechanical and Mechatronic Systems MMaMS 2014

*Keywords:* robot hand; modeling; grasping; neural networks; deep learning; object recognition; pose estimation

## 1. Introduction

Robotics is moving towards the research and development of technologies that allow the introduction of robots in our daily life. The optimal robot assistant should share a human environment and be able to cope with human presence and interact in a very friendly way. A number of problems need to be solved to create such applications, including transposing the movements used in everyday tasks, as well as finding out how to interpret human interactions and how to use all this knowledge to create robots that can successfully act as assistants.

* Corresponding author. *E-mail address:* pavol.bezak@stuba.sk


Peer-review under responsibility of organizing committee of the Modelling of Mechanical and Mechatronic Systems MMaMS 2014
doi:10.1016/j.proeng.2014.12.092

The need of having intelligent robots means that the complexity of programming must be greatly reduced, and robot autonomy must become much more natural. This challenge is relevant to a new generation of robots, which must interact with people, and operate in human environments.

There are many types of objects in the real world that can be subject to manipulation by human like hands. To ensure the robots have enough skills to interact with our environment we need to implement skills of the human hand. An autonomous robot hand will need to adapt to various grasp tasks in different situations. To solve such complex problem, artificial cognitive skills are needed to enable a robotic platform to take decisions for the execution of each specific task, and also to adapt to a human environment.

Robotic grasping is extremely difficult because of unknown objects and poses. Martinez and Collet made framework MOPED (Multiple Object Pose Estimation and Detection) in which they solved object recognition and pose estimation [1].

A robotic manipulator with a human like hand as its end-effector will be exceedingly effective in all its current and future applications. The multi-fingered hand was first introduced to reduce the limitation of conventional grippers and to increase the efficiency of a manipulator in executing grasping and manipulation task [2]. Salisbury [3] first successfully designed a human like robot hand, an advanced end-effector, for the purpose of grasping research. Since then, many robot hands were designed and used to perform tasks in industries, hazardous and remote places such as power plants and space applications. The UTAH MIT hand, the DLR hand and the Barrett hand are some examples which are widely used in industries and as grasping and manipulation research platforms [4, 5, 6]. This paper studies the modeling and contact control of the robot hand to improve efficiency in executing grasping tasks.

Grasp research concern to increase the capability of a robot hand to grasp objects of several shapes and stiffness with dexterity, by ensuring that no movement occurs between the object and the hand during contact. In whole finger manipulation, all links of a finger make contact with the object which brings more complexity in hand-object dynamics.

A number of problems such as hand modeling, hand-object contact modeling and control are prerequisite to be addressed to improve the efficiency and dexterity of the multifingered hand for grasping. The performance of the hand in grasping depends on the exact information of the object's properties, hand modeling, contact modeling between the hand and the object, grasp planning on the basis of the object location and contact point location between the object and the hand, contact force estimation and control of the hand according to hand-object dynamics. Considering the above issues, object grasping with a multifingered hand is presented in the following four steps: **1. Object Location**. The goal is to locate the object position by Visual Sensor. Object recognition, visual tracking and 3-D pose estimation are the advanced technology used in identifying objects in the environment. **2. Grasp Planning**. Plan the grasp based on the sensor data mounted on fingertip and motor. At this stage, it is necessary to define the number of contact points to grasp the object. Grasp is planned based on these contact points. Multiple contact points results in complex grasp planning. **3. Contact Modeling**. The contact model is essential to know the contact locations between the hand and the object, to estimate the contact force and to avoid any slippage due to the friction. **4. Grasp control**. Grasp control is complicated when the hand-object model is not known a-priori. Type and number of contacts are the factors in controlling grasp. The control task is carried out in 2 steps: At first, the hand arrives at the desired location and subsequently makes contact with the object and secondly it requires an ideal force to be applied on the object, depending on the object's stiffness, to grasp it without any slippage. In this paper, the three-fingered hand is selected for investigating object grasping by robot hand.

## 2. General problem formulation

The principal task in grasping involves the interaction between the object and the hand. Let's consider a fixed object with known coordinates and a finger of the robot hand. The task of grasping can be divided into two distinct problems. First task is to command the hand to reach the object location. Contact force is generated when the hand

hits the object, which needs to be dealt with. The contact force needs to be bounded to prevent any slippage or damage of the object.

**Multi-fingered hand kinematics**

The kinematic model of human hand according to [8] based on [9] is in Figure 1. For kinematic analysis the structure with 23 degrees of freedom is used.

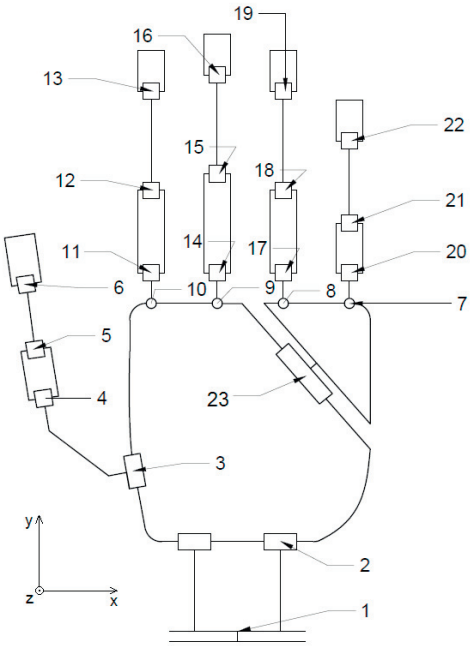


Fig. 1. Kinematic model of human hand [8]

*2.1. Forward Kinematic Model*

First of all, a coordinate system is assigned to each link of each finger k of the hand according to the standard Denavit-Hartenberg convention. Next, a set of Denavit-Hartenberg parameters are defined for each coordinate system. Using Matlab, the terms can be calculated.

*2.2. Inverse Kinematic Model*

Angles $\theta_{k2}$ and $\theta_{k3}$ can be calculated by taking into account that the two phalanxes of the finger are contained in a plane $k\Pi$.

Thereby, by applying the law of cosines to the BCD triangle, the following expression can be derived for the computation of the angle $\theta_{k3}$:

$$\theta_{k3} = \arccos\left(\frac{\left(\sqrt{{}^{k0}p_x^2 + {}^{k0}p_y^2} - A_1\right)^2 + {}^{k0}p_z^2 - A_2^2 - A_3^2}{2A_2A_3}\right) \tag{1}$$

Finally, the angle $\theta_{k2}$ is obtained from the difference between the angles $\alpha$ and $\beta$ in Fig. 3:

$$\theta_{k2} = \alpha - \beta = \arctan\left(\frac{{}^{k0}p_z}{\sqrt{{}^{k0}p_x^2 + {}^{k0}p_y^2} - A_1}\right) - \arctan\left(\frac{A_3\sin(\theta_{k3})}{A_2 + A_3\cos(\theta_{k3})}\right) \tag{2}$$

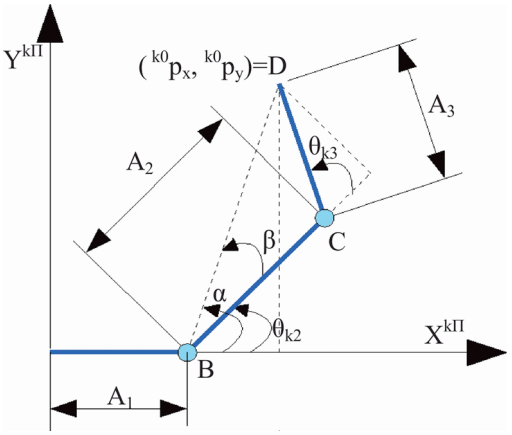


Fig. 2. Geometric relations of the links of a finger [7].

*2.3. Inverse dynamics model*

For each finger k, its inverse dynamic model can be represented by the following equation:

$$\tau_k = D_k(\Theta_k)\ddot{\Theta}_k + H_k(\Theta_k, \dot{\Theta}_k) + C_k(\Theta_k) \tag{3}$$

Where $\Theta_k$ is a $n_k \times 1$ vector with the joint values, $\tau_k$ is a $n_k \times 1$ vector with the joint torques, $D_k$ is the $n_k \times n_k$ inertia matrix, $H_k$ is the $n_k \times 1$ Coriolis-centrifugal vector and $C_k$ is the $n_k \times 1$ gravitational vector. $n_k$ is the number of joints of each finger $k(n_1 = n_2 = 3$ and $n_3 = 2)$.

## 3. CAD Model of Simplified Humanoid Robot Hand

After kinematic analysis the 3D model of simplified (three-fingered) humanoid hand can be created. For this task the software Autodesk Inventor was used. In the CAD model we did not assume actuators. The CAD model (Figure 3) serves us only for better imagination and for observation of possible motions and operations with hand.

It is possible to import the model from Autodesk Inventor to Matlab Simmechanics to further analysis. We used the tool smlink_linkinv. After successful import we gained also Simulink model of the model of the hand.

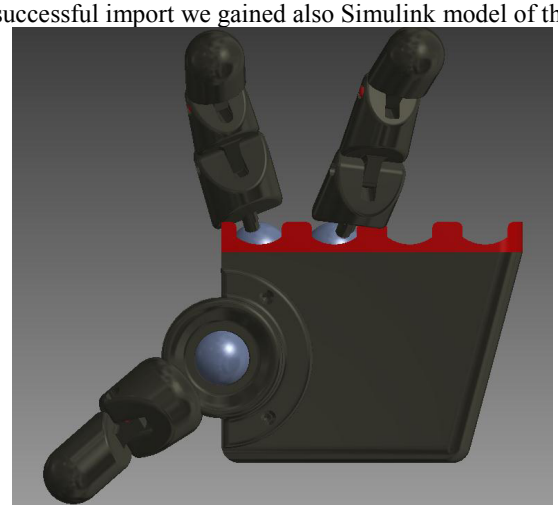

Fig. 3. CAD model of humanoid robot hand

In the Matlab Simulink model we can see parts of the simulated robotic hand after import from Autodesk Inventor (Figure 3). The model had to be edited to add the required functionality.

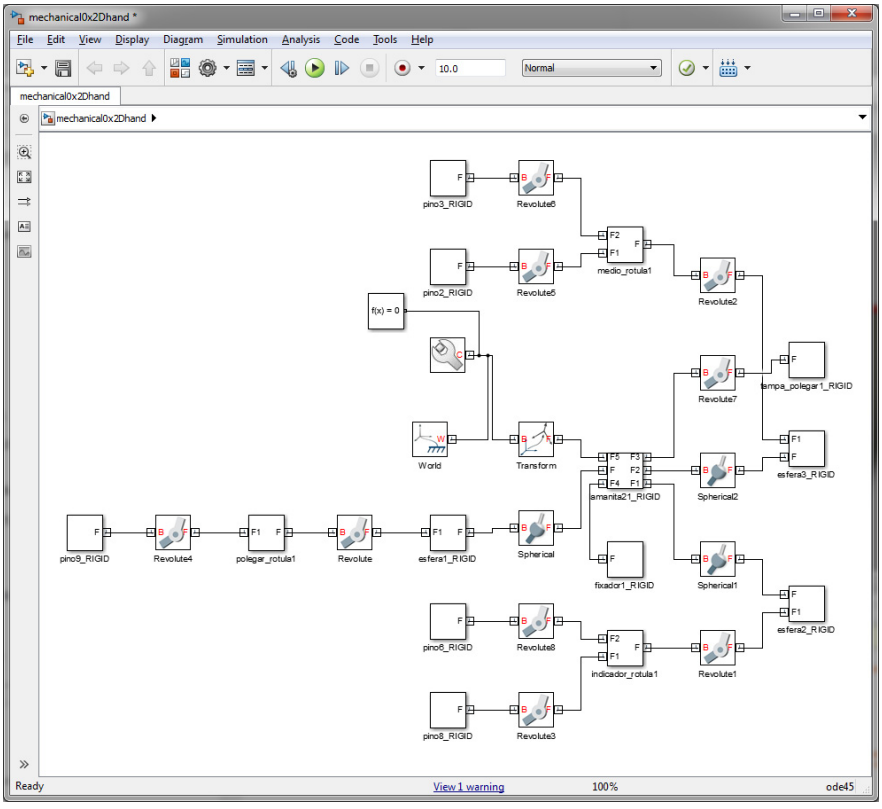


Fig. 4. Imported Matlab Simulink model of humanoid robot hand

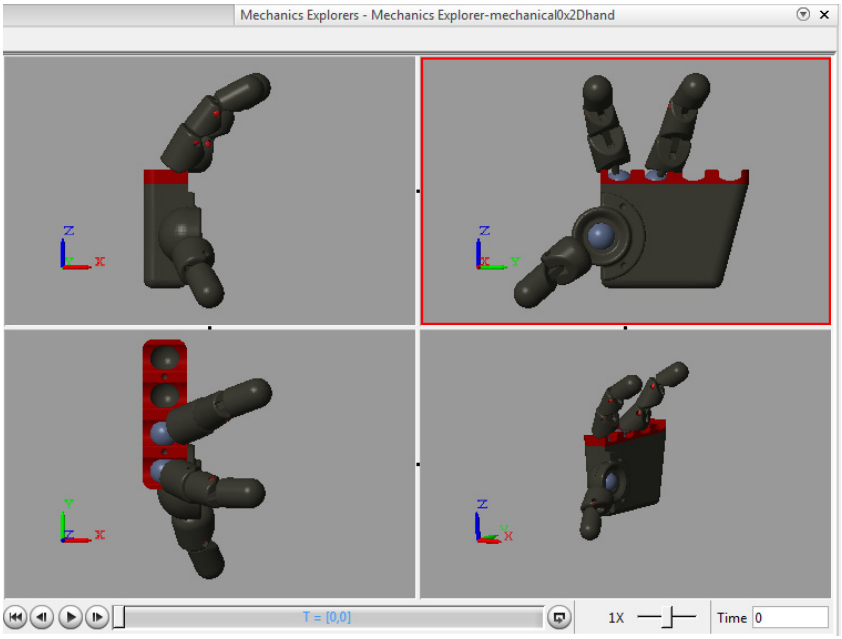


Fig. 5. 3D model of humanoid robot hand in Matlab Mechanics Explorer

## 4. Hand-arm control, object detection, recognition and pose estimation

During the execution of a manipulation task, unexpected or pre-planned interactions of the object with the environment may occur. The main goal of the control is to ensure that the robotic system does not lose the object and that the exchanged forces remain limited. A crucial point is the control of the contact forces between the object and the fingers. Keeping the contact forces within a certain range is important for several reasons. On the one hand, these forces must be sufficiently high to guarantee the satisfaction of the friction cone constraints; on the other hand, contact forces cannot be too high to avoid saturation of the motors and waste of energy, as well as to preserve the materials. In this context, the presence of the force sensors at the fingertips plays an important role for the control of both the arm and the hand [10].

Visual object detection is the most important step in robotic grasping. Many methods have been reported so far. As almost all the object recognition methods rely heavily on the accuracy of foreground object detection, efficient and reliable methods are necessary [11].

### *4.1. Brief introduction to artificial neural network*

ANN is a computational model that is inspired by the function of biological neural networks. ANN consists of a group of artificial neurons which processes information over interconnection [12, 15]. In spite of many versions of neural networks, all of them have similar characteristics. They have many neurons which are connected to many other neurons. The scheme of a typical neuron is shown in Fig. 6.

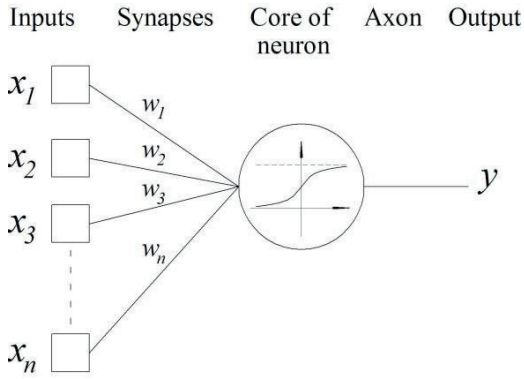


Fig. 6. The scheme of the neuron [12]

The artificial neuron has synapses which connect inputs of neuron to the core. The next it has the core which processes input signals. And also it has the axon which links to neurons of the next layer. Each synapse has a weight which defines how its input of the neuron influences on its state [12, 14].

State of a neuron is defined by following formula:

$$S = \sum_{i=1}^{n} x_i w_i \tag{3}$$

Where:

$n$–number of inputs of the neuron;
$x_i$–value of $i$-th neuron;
$W_i$–weight of $i$-th synapse;

The value of the axon is calculated by following formula:

$$Y = f(S) \tag{4}$$

where $f$ is an activation function. We use the Gaussian activation function, which is shown below:

$$f(x) = \frac{1}{1 + e^{-\alpha x}} \tag{5}$$

In machine learning and related fields, artificial neural networks (ANNs) are computational models inspired by an animal's central nervous systems (in particular the brain) which is capable of machine learning as well as pattern recognition. Artificial neural networks (Fig. 7) are generally presented as systems of interconnected "neurons" which can compute values from inputs.

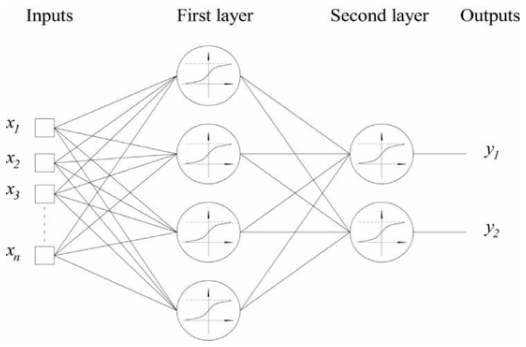


Fig. 7. The structure of the neural network [12]

### *4.2. Brief introduction to deep neural networks and deep learning*

In machine learning, a deep belief network (DBN) is a generative graphical model, or alternatively a type of deep neural network, composed of multiple layers of latent variables ("hidden units"), with connections between the layers but not between units within each layer. Deep learning is a set of algorithms in machine learning that attempt to model high-level abstractions in data by using model architectures composed of multiple non-linear transformations. Various deep learning architectures such as deep neural networks, convolutional deep neural networks, and deep belief networks have been applied to fields like computer vision, automatic speech recognition, natural language processing, and music/audio signal recognition where they have been shown to produce state-of-the-art results on various tasks.

### *4.3. Convolutional neural networks (CNN)*

A CNN is composed of one or more convolutional layers with fully connected layers (matching those in typical artificial neural networks) on top. It also uses tied weights and pooling layers. This architecture allows CNNs to take advantage of the 2D structure of input data. In comparison with other deep architectures, convolutional neural networks are starting to show superior results in both image and speech applications. They can also be trained with standard back propagation. CNNs are easier to train than other regular, deep, feed-forward neural networks and have many fewer parameters to estimate, making them a highly attractive architecture to use.

Comparing with the traditional object recognition based on the deep learning model, we focus on the object pose estimation including object recognition. Deep learning methods have the capability of recognizing or predicting large set of patterns by learning sparse features of small set of patterns. With this advantage, we can use a small set of poses to train the deep learning model, and then predict a large set of poses with the model [11].

For general objection recognition and image classification tasks, variants of Convolutional Neural Networks (CNNs) have emerged as robust supervised feature learning and classification tools, especially when combined with max-pooling (MPCNN) (Fig. 8) [13].

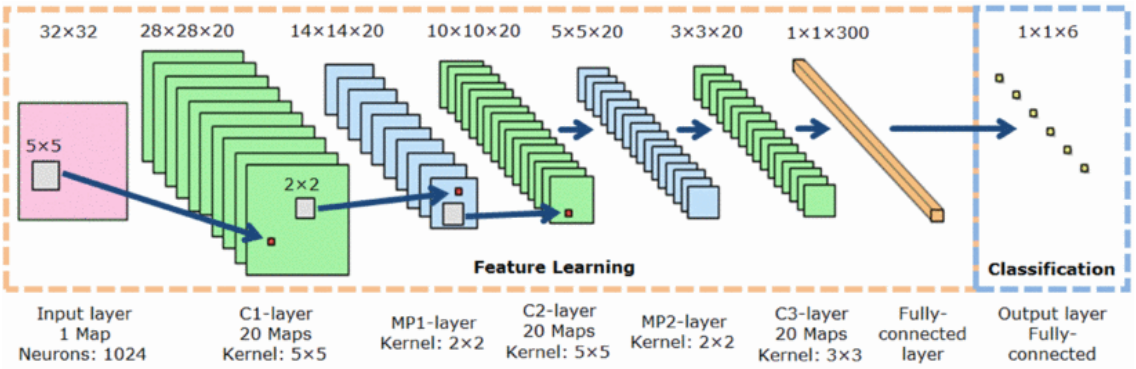


Fig. 8. MPCNN architecture using alternating convolutional and max-pooling layers [13]

MPCNNs include convolutional layers and subsampling layers. MPCNNs are different according to the variety of training and realization of convolutional and subsampling layers.

Convolutional layer

The parameters of the convolutional layer are: the number of maps, the size of the maps and kernel sizes. Each layer (L) includes maps (M). A kernel (K) of size is shifted over the valid region of the input image. Each map in Layer $L^n$ is connected to all maps in layer $L^{n-1}$. Neurons of a given map share their weights but have different input fields [13].

Max-pooling layer

The output of the max-pooling layer is determined by the maximum activation over non-overlapping rectangular regions. Max-pooling improves generalization performance [13].

Classification layer

To complete the MPCNN, a shallow Multi-layer Perceptron (MLP) is used. The output layer has one neuron per class in the classification task [13].

## 5. Experiment

### *5.1. Object Detection*

The system consists of the Matlab SimMechanics model of robotic hand scene with objects and simulated camera. The virtual objects are in the reach of the vision system and are recognized through the Matlab Computer Vision Toolbox with implemented model of MPCNN.

The input images come from RGBD camera data that opposed to simple 2D image data has been shown to significantly improve the grasp detection results.

Visual object detection is the first key action for robotic grasping. It is needed to use or develop reliable methods that effectively recognize foreground objects that are present at the input to the vision system and detect the objects from the background.

One of the possible approaches is to use sparse coding or K-means clustering. The first step is to build dictionary of objects. Clustering enables us to separate groups of color components of the images with background and objects. Each component has a location in feature space and it is important to find partitions such that components within each cluster are as close to each other as possible and as far from components in other clusters as possible [11].

### 5.2. Object Recognition and Pose Estimation

Using the simulation, we implemented object pose estimation and pose estimation using the methods of deep learning, which have the capability of recognizing or predicting large set of patterns by learning sparse features of small set of patterns. With this advantage, we can use a small set of poses to train the deep learning model, and then predict a large set of poses with the model [11].

### 5.3. Robotic Grasping

This stage presents the system that changes the position and orientation of the gripper in order to grasp the objects.

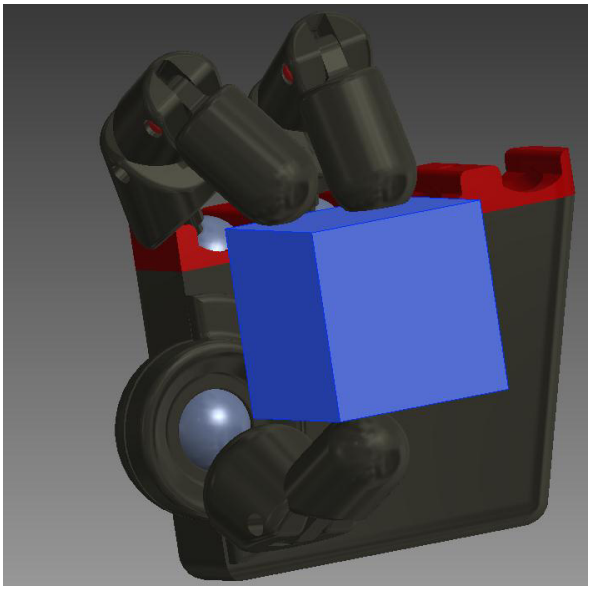

Fig. 9. 3D model of humanoid robot hand grasping the object

## Conclusion

The developed models described in previous sections have been implemented in selected Matlab Toolboxes (Computer Vision System Toolbox, Deep Learning Toolbox and SimMechanics).

This paper represents a simulated model of a multi-fingered robotic hand for grasping tasks. The model includes kinematics, dynamics, object representation and contact modelling. The contact model determinates the forces that are applied to the object by the robotic hand. The object detection, object recognition and robotic hand pose estimation are based on Max-pooling Convolutional Neural Networks – one of the most popular deep learning models. Using deep learning enables to avoid hand-engineering features, learning them instead.

## References

[1] A. Collet, M. Martinez, S. S. Srinivasa, The MOPED framework: Object Recognition and Pose Estimation for Manipulation, in International Journal of Robotics Research, 2011.

[2] R. Hasan, A. Rahideh, H. Shaheed, Modeling and interactional control of the multifingered hand, 19th International Conference on Automation and Computing (ICAC), pp. 6-10, 13-14 Sept. 2013.

[3] Ch. Pellerin, The Salisbury Hand, Industrial Robot: An International Journal, Vol. 18(1991) Issue: 4, pp. 25-26.

[4] W.T. Townsend, The Barrett Hand grasper-programmably flexible part handling and assembly, Industrial Robot: An International Journal, Vol. 10(3), 2000, pp. 181-188.
[5] M. Grebenstein, The DLR hand arm system, IEEE International Conference on Robotics and Automation (ICRA), 2011, pp. 3175-3182.
[6] S. Jacobsen, Design of the Utah/M.I.T. dexterous hand, IEEE International Conference on Robotics and Automation (ICRA), Vol.3, 1986, pp. 1520-1532.
[7] J.A. Corrales, C.A. Jara, F. Torres, Modelling and simulation of a multi-fingered robotic hand for grasping tasks, 11th International Conference on Control Automation Robotics & Vision (ICCARV), vol., no., pp.1577-1582, 2010.
[8] I. Virgala, M. Kelemen, S. Mrkva, Kinematic Analysis of Humanoid Robot Hand, American Journal of Mechanical Engineering 1.7 (2013): pp. 443-446.
[9] S.M. Lee, K.D. Lee, H.K. Min, T.S. Noh, J.W. Lee, Kinematics of the Robomec robot hand with planar and spherical four bar linkages for power grasping, IEEE International Conference on Automation Science and Engineering (CASE), pp. 1120-1125, 2012.
[10] F. Ficuciello, L.Villani, Compliant hand-arm control with soft fingers and force sensing for human-robot interaction, 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob), pp.1961-1966, 2012.
[11] Y. Jincheng, K. Weng, G. Liang, G. Xie, A vision-based robotic grasping system using deep learning for 3D object recognition and pose estimation, IEEE International Conference on Robotics and Biomimetics (ROBIO), pp.1175-1180, 2013.
[12] Y. Karavaev, A. Klekovkin., P. Bezak, The implementation of microprocessor device for drilling process monitoring based on artificial neural network, International Conference on Process Control, pp.163-167, 2013.
[13] J. Nagi, F. Ducatelle, G.A. Di Caro, D. Ciresan, U. Meier, A. Giusti, F. Nagi, J. Schmidhuber, L.M. Gambardella, Max-pooling convolutional neural networks for vision-based hand gesture recognition, IEEE International Conference on Signal and Image Processing Applications (ICSIPA), pp.342-347, 2011.
[14] H. Mekki, M. Chtourou, Variable structure neural networks for adaptive robust control using evolutionary artificial potential fields, Journal of Electrical Engineering, Vol. 64, No. 1, 2013, 3–11
[15] A. Akdagli, A. Toktas, A. Kayabasi, I. Develi, An application of artificial neural network to compute the resonant frequency of e-shaped compact microstrip antennas, Journal of Electrical Engineering, Vol. 64, No. 5, 2013, 317–322