

**Роботика и флексибилна  
аутоматизација**

**Robotics  
and flexible automation**

# Robotized Acquisition of Data for Characterisation and Calibration of Soft Fingertip Sensor

Mirko Raković, *member, IEEE*, Srdjan Savić, *member, IEEE*, Darko Todorović, Miroslav Božić, Andrej Čilag, Vladimir Sibinović, Branislav Borovac, *member, IEEE*

**Abstract**—The tactile sensory system is of high importance for robotic grasping and manipulation and provides the robot with information about properties of physical interaction with the object. Out of many properties that can be measured, contact forces are of the greatest importance. This information can be used to determine the quality of the grasp and it can be used as a force feedback for the control of the robot's hand and arm.

As a part of the development of robotic hand we are designing a fingertip sensor that can sense a contact force vector and the location of the contact point. The sensor is distributed, with a matrix of magnets and magnetic sensors, and covered with silicone layer. In order to characterise and calibrate each sensor a procedure for acquiring the data is extensive and requires a large amount of variations of contacts between external object and sensor. This paper presents a systematic way to conduct the repeatable process of data acquisition with industrial robot that covers variations in position, orientation and intensity of the force vector at the contact point.

**Index Terms**—force sensors, soft computing with sensor data, robotics.

## I. INTRODUCTION

HUMANS interact with the environment using rich sensory information. To perceive the environment we use vision, proprioception and tactile sensing almost without realizing it. A man-made environment is challenging for a robot due to the constant change in type and placement of objects it needs to interact with and manipulate to accomplish the task. Also, objects in human's surrounding are mainly designed to adapt to capabilities of human's hands. When manipulating these objects humans are heavily relying on the feedback from tactile sensing in order to hold object properly, to avoiding slippage, and to feel its properties (texture, temperature, stiffness etc.). Apart from the ability to detect and recognize objects, a robot should have some general object grasping and manipulation capabilities allowing for physical interaction. Thus, one of important tasks of robots operating in human-centric environment is autonomous dexterous object manipulation. It is clear that tactile sensory system is of paramount importance and we are still lacking sensor systems that can provide information that a human receives through fingertips.

To address this problem we are designing a sensor that can detect the location of the contact and estimate the vector

of the contact force. In this paper we present process for acquiring data for characterization and calibration of a Hall effect based tactile sensor that should estimate the force vector data and location of the contact at the sensor's surface. The transduction technology is magnetic. Structure of sensor consists of magnets positioned inside a silicon layer placed above the PCB with Hall effect sensors. The silicon is also a final surface of the sensor body and thus provides a soft cover which enables adaptation to the shape of objects in contact. Due to complex shape of magnetic field when silicon body is deformed and inability to precisely position magnets and Hall sensors, bubbles of air trapped inside the silicon each sensor needs to be calibrated and characterised. In order to have a process that is repeatable and that can cover the whole surface of the sensor with different directions of force vector and its intensity we are proposing a setup for acquiring data for calibration and characterisation. Through this process each sensor is pressed around 260.000 times and data on position and orientation of the contact, force/torque vector from reference sensor, and output from eight three axis Hall sensors is acquired.

The paper is organized as follows. In Section II we provide review of related tactile sensors. Section III presents a description of the sensor design. Section IV shows the experimental methods used to acquire data for characterization and calibration of the sensor and Section V describes the program for controlling industrial robotic arm. Finally, Section VI draws conclusions and outlines the future work.

## II. RELATED WORK

Tactile sensing is an important modality within robot perception system, which has a fundamental role in physical interaction, including the tasks of in-hand manipulation [1] and active exploration of object's properties [2], [3]. It is particularly important in the context of collaborative and social robotics, to provide safe and intuitive physical human-robot interaction [4], [5]. Tactile feedback is a necessary prerequisite for multiple tasks: to provide and maintain a stable grasp, to detect object slippage, to detect and manage physical interaction with human coworkers, etc. Together with other perception modalities, especially vision, tactile sensors increase robots' autonomy and enable them to operate in dynamic unstructured environments and to manipulate unknown objects.

Tactile sensors may be characterized by multiple features. Selection of the proper tactile sensor and its design specifications is task-dependent. According to [6], the following design

M. Raković, S. Savić, A. Čilag and B. Borovac are with the Faculty of Technical Sciences, University of Novi Sad, Serbia. e-mail: rakovicm@uns.ac.rs  
D. Todorović and V. Sibinović are with the Faculty of Electronic Engineering, University of Niš, Serbia.  
M. Božić is with the High Tech Engineering Center, Belgrade, Serbia.

criteria have been recognized as relevant for tactile sensors: spatial resolution, sensitivity, hysteresis, frequency response, wiring, flexibility, surface properties and robustness. Some of these criteria are related and coupled and therefore most of the tasks require certain compromises to be made. Higher spatial resolution is desirable for active exploration and in-hand manipulation. However it increases the signal acquisition time from multiple sensor cells and imposes the problem with massive wiring [7]. Therefore, it imposes certain limitations for other tasks like slippage and vibration detection, surface texture recognition and reactive control which require high sensitivity and high frequency response. Moreover, sensitivity of tactile sensor, defined as the smallest detectable variation in force, is inversely-proportional to the dynamic range of measurement, imposing another compromise. The use of soft materials (silicone or rubber) as viscoelastic layers around rigid core sensor elements is very common and it helps to increase surface friction and to provide desired compliance of the fingertip. However, it limits the frequency bandwidth of the sensor [7].

Tactile sensors may be classified under multiple criteria. They can be unimodal, measuring only contact force, or multimodal, including measurement of other physical entities like temperature, pressure, proximity, etc. Multimodal sensors may provide richer information about interaction properties. Regarding the unimodal tactile sensors, most important classification criterion is based on type of the transduction technology used. Let us mention just a few state-of-the-art sensors, based on different transduction technologies: piezoresistive [8], [9], capacitive [10], [11], piezoelectric [12], optical [13], [14], magnetic [15], etc. Tactile sensors can measure various properties. There are (i) 1-axis sensors, measuring only normal component of contact force, (ii) 3-axis sensors, measuring both normal and shear components and (iii) 6-axis force/torque sensors. Also, some sensors can measure or estimate location of contact point, surface texture, temperature, etc. Tactile sensors may consist of a single measuring element, or they can be distributed in form of arrays or matrices of measuring elements, so called cells. However, sensor integration into a robotic hand imposes multiple constraints and severe requirements, regarding available space, wiring and high functional demands. Only few of the conceptual solutions are currently at the required technology readiness level to be used in robotics. Some of the successful applications of tactile sensors in humanoid robotics include: iCub [15], DLR hand [16], Gifu hand II [17] and IH2 hand [18]. So far, several commercially available tactile sensors for robotic applications have also appeared on the market. The most advanced available sensor is the BioTac fingertip from Syntouch [19].

We do not aim to give a complete overview of force sensing technologies and their characterization methods, but rather to point out related work relevant for our paper. For a systematic and complete survey of tactile sensing technologies we refer to [6], [20]. We primarily focus our survey on the 3-axis force sensors based on magnetic transductive technology, which is in line with our proposed concept. In [21], [22] a low-cost 3-axis tactile sensor, based on the Hall effect, is presented. The sensor consists of Hall effect sensor microchip,

covered with soft elastomer layer, with a permanent magnet above the sensor. These papers report conceptually similar solution, as the one presented in this paper. However, the sensor described in [21], [22] use a single Hall effect sensor aiming primarily on high sensitivity, low hysteresis and good repeatability, while our approach includes a matrix of Hall effect cells, thus providing better spatial resolution and wider possibilities for estimation of the contact point location. In our previous work, we have reported on the prototype version of the magnetic tactile sensor with 2x2 matrix of cells, designed for the fingertip segment of an anthropomorphic robot hand [23], [24]. Slightly altered version of sensor presented in [21], [22], with a flexible textile cover over the silicon layer, has been integrated into the hand of iCub humanoid robot [25]. Regarding the characterization procedure, in [21] the setup for data acquisition included an Arduino Leonardo board, a commercial semi-spherical 3-axis optoforce sensor from OptoForce and a motorized micro positioning system, while in [22] the setup included a Cartesian robot, a 6-axis F/T sensor (ATI nano 17) mounted on the robot end-effector, and a data acquisition board (NI DAQ USB-6008). However, in our approach for data acquisition within sensor characterization process we used ABB IRB140 anthropomorphic robot arm, which enabled us to apply forces to the sensor in various spatial directions, which is not possible with Cartesian robot configuration. Also, we attached the reference 6-axis F/T sensor to the ground, below the custom 3-axis tactile sensor, instead of putting it between the robot arm and end-effector. In [21], [22] sensor characterization curve was obtained by interpolating the measurement values using linear regression with a quadratic model. The solution proposed in [21], [22] has been generalized and the sensor called uSkin has been presented in [15], [26]. uSkin enables distributed perception with a 4x4 matrix of sensor cells which provides higher spatial density. In [15] three different calibration methods have been used and compared for the uSkin sensor: linear regressions with and without quadratic model and a feed-forward neural network (FNN) with 1 hidden layer. Calibration method based on neural networks was also used in [27] to calibrate a distributed tactile sensor with optoelectronic measuring cells. The authors used feed-forward neural network with 6 hidden layers, each containing 30 neurons, a sigmoidal activation function, and the output layer with 6 neurons and linear activation function.

### III. DESIGN OF THE SENSOR

#### A. Sensor Design and Working Principle

As previously mentioned, the measuring element is a 3D linear Hall effect sensor. The sensor changes the output voltage with respect to the intensity and direction of magnetic field that passes through sensor. Sensing elements are placed on the top side of PCB (Fig. 1), whereas on the bottom side is an ARM Cortex-M4 MCU with floating point unit, running at 84MHz.

The board is manufactured in three variations, with the distance between sensors in both, x and y direction set at 5mm, 7.5mm and 10mm (Fig. 2). This variation is made so that we can experimentally validate how spacing between Hall effect

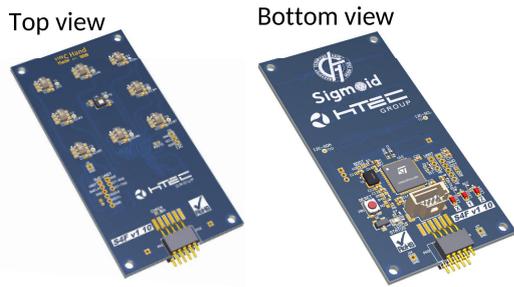


Fig. 1. PCB design of tactile sensor with Hall sensor on top and MCU on bottom side

sensors influences the resolution and sensitivity of intensity and location of the contact force. In this stage, the neodymium disc magnets of 2mm in radius and 1mm thickness are placed in a 3x3 matrix.

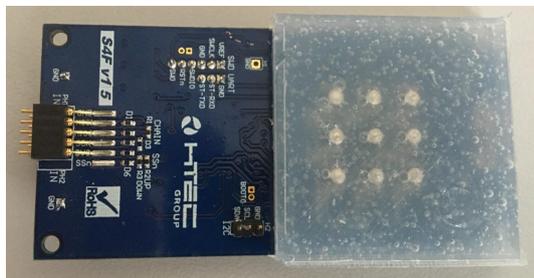


Fig. 2. The prototype of sensor with silicon layer and 3x3 matrix of magnets with distance of 5mm in x and y direction

### B. Simulation of Magnetic Field

To better understand the distribution of magnetic field in our matrix of magnets we turned to simulations. By modeling the permanent magnet and its magnetic field, we could test different combinations of spacing and orientation to find the best solution for experimental testing.

Using [28] as a starting point for modeling, we have recreated our environment, where we are using cylindrical magnets with axial polarization that have a diameter of 2mm and a height of 1mm. We are using these magnets in a 3x3 matrix 2mm above from the sensor plane. For easier presentation the results are shown just for one row of 3 magnets. Figure 3 shows the results of the simulation, where on the top the magnets are all orientated the same, and on the bottom view the center magnet has a different orientation from the two on the sides. At this point, based on the simulation, we have decided to orient magnets all the same way. With this configuration we have a fairly uniform distribution at the region of interest, i.e. at the sensor plane.

In the simulation program we can change the geometry and show the results when a force displaces the magnet from its original position. Figure 4 shows the magnetic field when the displacement of the central magnet is 0.5mm from starting position. This is shown on top, while on the bottom part we have a displacement of 0.1mm with a rotation of 5 degrees. This simulates a force that is not applied perpendicular to the

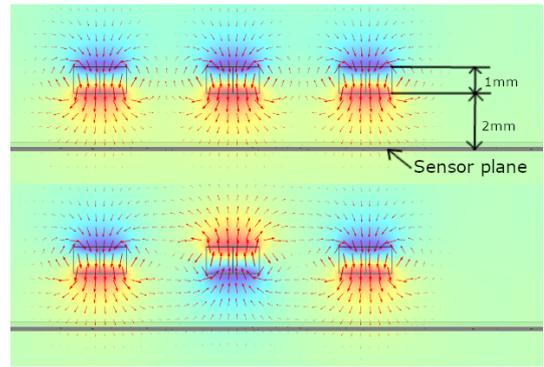


Fig. 3. Simulation results for three magnets with the same orientation and when center magnet is differently orientated, at a spacing of 5mm

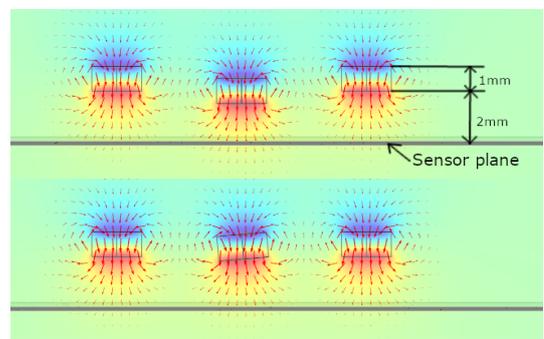


Fig. 4. Simulation results for displacement, top - 0.5mm displacement, bottom - 0.1mm and 5 degrees rotation

sensor, and will be something that we will investigate further in the future.

### IV. DATA ACQUISITION SETUP

Hardware of the acquisition setup consists of ABB IRB140 industrial robotic arm, tactile sensor, reference 6 axis force/torque sensor and the PC (Fig. 5). All three parts of the system are connected with PC through RS232 communication protocol.

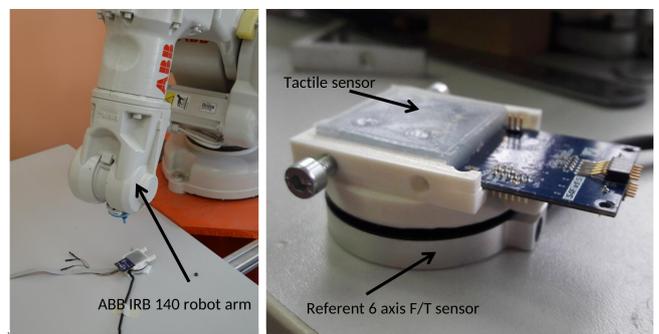


Fig. 5. Experimental setup with industrial robot, fingertip sensor and reference 6 axis force torque sensor

We have developed an application for acquiring the data from all three data source. The application waits for the data from industrial robot that is being sent when the robot precisely presses the tactile sensor. The data structure from

industrial robot contains all the information about the position and orientation of the tool with respect to the sensor. After receiving this data, application sends request to referent F/T sensor that contains precise measurements on force and torque. Afterwards, the application sends the request for measurement to tactile sensor. The structure received from tactile sensor contains 8-bit or 12-bit (depending on the configuration) value that corresponds to the intensity of magnetic field for three directions, from all eight Hall sensors. This set of data represents one measurement that is repeated for each change in the position and orientation of pressing tool i.e. industrial robot arm.

## V. CONTROL OF INDUSTRIAL ROBOT

For controlling the robotic arm, the program has been written in RAPID programming language, and RobotStudio software has been used for developing the simulation. RobotStudio environment enabled fast generation of appropriate coordinate systems, efficient checking of the accessibility of targets and paths, as well as the possibility of offline testing.

RAPID program is organized in two modules: The calibration module (*CalibData*) in which the coordinate systems of the environment are located, as well as the parameters of the tools and their coordinate systems; and a module (*MainModule*) which contains the main procedure and a robot control algorithm. The task of the *Main* procedure (see Algorithm 1) is to execute the pressing algorithm (see Algorithm 2), and to print out the number of iterations performed on the screen. *Pressure\_sensing\_algorithm* is used for pressing the sensor's silicon layer properly - on the proper points, with specified orientation.

Described by Algorithm 2,  $x$  and  $y$  coordinates of the first pressing point are calculated (see Fig. 6). After that, the robot arm positions its Tool Center Point (TCP) above the calculated point, so the  $z$  axis of TCP is perpendicular to the silicon layer surface of the sensor. The robot then performs pressing the silicon layer in  $dz$  mm increments, to 2mm depth, where  $dz = 0.1mm$ .

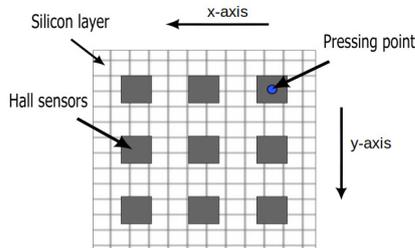


Fig. 6. First pressing point on silicon layer

After pressing the first point, the robot arm changes its orientation around the  $x$  and the  $z$ -axis, to get data from multiple orientations of the contact force. This change is implemented within the RAPID program using nested FOR loops. Change of the orientation of TCP around the pressing point on the sensor plane inside the silicon layer is done in equidistant increments.

---

### Algorithm 1: Main()

---

**Data:** Call for *Pressure\_sensing\_algorithm* procedure;  
**Result:** The number of *Pressure\_sensing\_algorithm* execution;  
initialization;  
**while true do**  
     $count := count + 1$ ;  
    print  $count$ ;  
    *Pressure\_sensing\_algorithm*();  
**end**

---



---

### Algorithm 2: *Pressure\_sensing\_algorithm*()

---

**Data:** Position and orientation offsets for pressing point;  
**Result:** Position and orientation of pressing point;  
initialization;  
calculate  $first\_pressing\_point$ ;  
 $d\vartheta := \vartheta_{max}/n$ ;  
 $d\varphi := 360^\circ/m$ ;  
**while  $x \leq sensor\_distance\_x$  do**  
    **while  $y \leq sensor\_distance\_y$  do**  
        pressing with  $dz$  increments;  
        **for  $angle\_z$  from  $0^\circ$  to  $200^\circ$  by  $d\varphi$  do**  
            **for  $angle\_x$  from  $d\vartheta$  to  $\vartheta_{max}$  by  $d\vartheta$  do**  
                pressing with  $dz$  increments;  
            **end**  
        **end**  
        **for  $angle\_z$  from  $200^\circ$  to  $360^\circ$  by  $d\varphi$  do**  
            **for  $angle\_x$  from  $d\vartheta$  to  $\vartheta_{max}$  by  $d\vartheta$  do**  
                pressing with  $dz$  increments;  
            **end**  
        **end**  
        reset  $point.orientation$ ;  
         $point.y := point.y + dy$ ;  
    **end**  
     $point.x := point.x + dx$ ;  
    reset  $point.y$ ;  
**end**

---

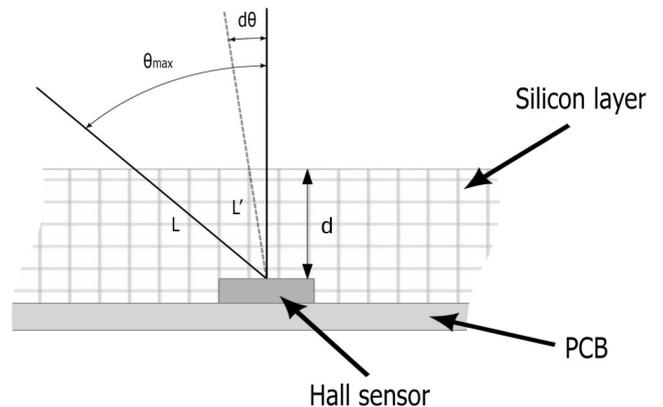


Fig. 7. Calculating new pressing point, after angle change

The first rotation is performed around the  $x$ -axis (angle  $\vartheta$ ). Change of the angle  $\vartheta_{\max}$  takes place in  $\mathbf{n}$  iterations, with each iteration being in steps of  $d\vartheta$  degrees, and  $\vartheta_{\max} = 45^\circ$  and  $n = 6$ :

$$d\vartheta = \vartheta_{\max}/n.$$

Due to the change of the pressing angle on the silicone layer, made around a point on the sensor surface inside the layer, the distance of the robotic arm between the two points differentiates, depending on the angle of rotation. As shown in Fig. 7, the distance between the points on the silicone layer and the surface of the sensor is determined as follows:

$$L = d/\cos(\alpha_x),$$

where  $d$  is silicon layer thickness,  $L$  represents the distance from the point on the surface of the silicon layer and the point on the surface of the sensor, and the  $\alpha_x$  represents the rotation around the  $x$ -axis. After calculating the required distance, a point on the silicone layer is found offset from the point on the sensor surface, along the  $z$ -axis of the TCP, ensued by pressing of  $2\text{mm}$ . After the final set  $\vartheta$  angle is reached, the orientation changes around the  $z$ -axis. The change of orientation of  $360^\circ$  around the  $z$ -axis takes place in  $\mathbf{m}$  iterations ( $m = 10$ ), in steps of  $d\varphi$  degrees:

$$d\varphi = 360^\circ/m.$$

The  $360^\circ$  rotation is performed in two passes due to the limited range of motion of the individual joints of the robotic arm. It is not possible to rotate from  $0^\circ$  to  $360^\circ$  in one direction of rotation. After rotation of  $200^\circ$ , the orientation returns to the beginning, which allows the joints to perform the remaining  $160^\circ$ .

The position of the pressing point is moved along the  $x$  and  $y$ -axes, in  $\mathbf{dx}$  and  $\mathbf{dy} = 5\text{mm}$  increments, after the rotation around  $x$  and  $z$ -axes, for a given point, is completed. The translation movement of the point makes a grid, size  $5 \times 5 = 25$  points of contact in total. At each point, the rotation around  $x$  and  $z$ -axes is performed. After processing all 25 points, the algorithm is repeated to achieve a larger number of data.

## VI. CONCLUSION

The paper described the hardware and software setup for automatic acquisition of large amount of data needed for characterization and calibration of tactile sensor. To autonomously and repeatedly perform this process, we used industrial robotic arm with the pressing tool as an end-effector. The developed application acquires data from: (i) robot controller on the location and orientation of the contact, (ii) force and torque values from 6 axis referent F/T sensor and (iii) 24 signals from eight 3D linear Hall sensors. These data will be used to find the relationship between outputs from Hall sensors on one side, and contact point location and vector of force at the contact point on other side. For this problem we will consider different approaches and it will be a topic for our future work.

## ACKNOWLEDGMENT

This work is supported by Innovation fund of Serbia under the project "Three-Finger HRC Robotic Hand". We would also like to thank to all engineers who were involved in the development of firmware and manufacturing of PCB.

## REFERENCES

- [1] Raul Fernandez, Andres S. Vazquez, Ismael Payo, and Antonio Adan, A Comparison of Tactile Sensors for In-Hand Object Location, Journal of Sensors, Hindawi Publishing Cooperation, pp. 1-12, Vol. 2016, (2016)
- [2] Seminara L, Gastaldo P, Watt SJ, Valyear KF, Zuher F and Mastrogiovanni F (2019) Active Haptic Perception in Robots: A Review. Front. Neurobot. 13:53. doi: 10.3389/fnbot.2019.00053
- [3] U. Martinez-Hernandez, T. J. Dodd and T. J. Prescott, "Feeling the Shape: Active Exploration Behaviors for Object Recognition With a Robotic Hand," in IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 48, no. 12, pp. 2339-2348, Dec. 2018, doi: 10.1109/TSMC.2017.2732952.
- [4] Andrea Cirillo, Pasquale Cirillo, Giuseppe De Maria, Ciro Natale, and Salvatore Pirozzi, A Distributed Tactile Sensor for Intuitive Human-Robot Interfacing, Journal of Sensors, Hindawi Publishing Cooperation, Vol. 2017,
- [5] Avelino, J., Paulino, T., Cardoso, C., Nunes, R., Moreno, P., Bernardino, A. (2018). Towards natural handshakes for social robots: human-aware hand grasps using tactile sensors, Paladyn, Journal of Behavioral Robotics, 9(1), 221-234. doi: <https://doi.org/10.1515/pjbr-2018-0017>
- [6] Zhanat Kappassov, Juan Antonio Corrales Ramon, Véronique Perdereau. Tactile sensing in dexterous robot hands - Review. Robotics and Autonomous Systems, Elsevier, 2015, 74, Part A, pp.195-220.10.1016/j.robot.2015.07.015. hal-01680649
- [7] M. Cutkosky, R. Howe, W. Provancher, Force and tactile sensors, in: B. Siciliano, O. Khatib (Eds.), Springer Handbook of Robotics, Springer Berlin Heidelberg, 2008, pp. 455-476. doi:10.1007/978-3-540-30301-5\_20.
- [8] Pyo, S., Lee, J., Kim, M. et al. Polymer-based flexible and multi-directional tactile sensor with multiple NiCr piezoresistors. Micro and Nano Syst Lett 7, 5 (2019). <https://doi.org/10.1186/s40486-019-0085-6>
- [9] Youzhi Zhang, Jinhua Ye, Zhengkang Lin, Shuheng Huang, Haomiao Wang and Haibin Wu, A Piezoresistive Tactile Sensor for a Large Area Employing Neural Network, Sensors 2019, 19(1), 27; <https://doi.org/10.3390/s19010027>
- [10] Z. Ji, H. Zhu, H. Liu, T. Chen and L. Sun, "A flexible capacitive tactile sensor for robot skin," 2016 International Conference on Advanced Robotics and Mechatronics (ICARM), Macau, 2016, pp. 207-212, doi: 10.1109/ICARM.2016.7606920.
- [11] A. Schmitz, M. Maggiali, L. Natale, B. Bonino and G. Metta, "A tactile sensor for the fingertips of the humanoid robot iCub," 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, 2010, pp. 2212-2217, doi: 10.1109/IROS.2010.5648838.
- [12] S. Kim, H. Shin, K. Song and Y. Cha, "Flexible piezoelectric sensor array for touch sensing of robot hand," 2019 16th International Conference on Ubiquitous Robots (UR), Jeju, Korea (South), 2019, pp. 21-25, doi: 10.1109/URAI.2019.8768644.
- [13] M. Lambeta et al., "DIGIT: A Novel Design for a Low-Cost Compact High-Resolution Tactile Sensor With Application to In-Hand Manipulation," in IEEE Robotics and Automation Letters, vol. 5, no. 3, pp. 3838-3845, July 2020, doi: 10.1109/LRA.2020.2977257.
- [14] Wenzhen Yuan, Siyuan Dong and Edward H. Adelson, GelSight: High-Resolution Robot Tactile Sensors for Estimating Geometry and Force, Sensors 2017, 17(12), 2762; <https://doi.org/10.3390/s17122762>
- [15] T. P. Tomo et al., "A New Silicone Structure for uSkin—A Soft, Distributed, Digital 3-Axis Skin Sensor and Its Integration on the Humanoid Robot iCub," in IEEE Robotics and Automation Letters, vol. 3, no. 3, pp. 2584-2591, July 2018, doi: 10.1109/LRA.2018.2812915.
- [16] M. W. Strohmayer, H. P. Saal, A. H. Potdar and P. van der Smagt, "The DLR touch sensor I: A flexible tactile sensor for robotic hands based on a crossed-wire approach," 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, 2010, pp. 897-903, doi: 10.1109/IROS.2010.5650191.
- [17] H. Kawasaki, T. Komatsu and K. Uchiyama, "Dexterous anthropomorphic robot hand with distributed tactile sensor: Gifu hand II," in IEEE/ASME Transactions on Mechatronics, vol. 7, no. 3, pp. 296-303, Sept. 2002, doi: 10.1109/TMECH.2002.802720.
- [18] <https://www.prensilia.com/portfolio/ih2-azzurra/> - accessed 2020-8-10.

- [19] <https://syntouchinc.com/robotics/> - accessed 2020-8-10.
- [20] R. S. Dahiya, G. Metta, M. Valle and G. Sandini, "Tactile Sensing—From Humans to Humanoids," in *IEEE Transactions on Robotics*, vol. 26, no. 1, pp. 1-20, Feb. 2010, doi: 10.1109/TRO.2009.2033627.
- [21] T. Paulino et al., "Low-cost 3-axis soft tactile sensors for the human-friendly robot Vizzy," 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 2017, pp. 966-971, doi: 10.1109/ICRA.2017.7989118.
- [22] L. Jamone, L. Natale, G. Metta and G. Sandini, "Highly Sensitive Soft Tactile Sensors for an Anthropomorphic Robotic Hand," in *IEEE Sensors Journal*, vol. 15, no. 8, pp. 4226-4233, Aug. 2015, doi: 10.1109/JSEN.2015.2417759.
- [23] Savić S., Raković M., Penčić M., Nikolić M., Dudić S., Borovac B.: Design of an Underactuated Adaptive Robotic Hand with Force Sensing , 3. International Conference on Electrical, Electronic and Computing Engineering (IcETRAN), Zlatibor: ETRAN Society, 13-16 Jun, 2016
- [24] Raković M., Beronja M., Batinica A., Nikolić M., Borovac B. (2017) 3-Axis Contact Force Fingertip Sensor Based on Hall Effect Sensor. In: Rodić A., Borangiu T. (eds) *Advances in Robot Design and Intelligent Control*. RAAD 2016. *Advances in Intelligent Systems and Computing*, vol 540. Springer, Cham. [https://doi.org/10.1007/978-3-319-49058-8\\_10](https://doi.org/10.1007/978-3-319-49058-8_10)
- [25] A. C. Holgado et al., "Magnetic 3-axis Soft and Sensitive Fingertip Sensors Integration for the iCub Humanoid Robot," 2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids), Toronto, ON, Canada, 2019, pp. 1-8, doi: 10.1109/Humanoids43949.2019.9035062.
- [26] Tito Pradhono Tomo, Sophon Somlor, Alexander Schmitz, Lorenzo Jamone, Weijie Huang, Harris Kristanto and Shigeki Sugano, Design and Characterization of a Three-Axis Hall Effect-Based Soft Skin Sensor, *Sensors* 2016, 16(4), 491; <https://doi.org/10.3390/s16040491>
- [27] Costanzo, M.; De Maria, G.; Natale, C.; Pirozzi, S. Design and Calibration of a Force/Tactile Sensor for Dexterous Manipulation. *Sensors* 2019, 19, 966.
- [28] A.M. Vucković, S. Aleksić, *Magnetic Field Determination for Different Block Permanent Magnet Systems*, FACTA UNIVERSITATIS (NIS) SER.: ELEC. ENERG. vol. 23, no. 3, Decmber 2010, 259-272

# Four-bar Linkage Mechanism Optimization for Linkage Driven Underactuated Robotic Finger

Lazar Matijašević, *PhD Student*, Petar B. Petrović, *Full Professor*

**Abstract** — When designing linkage driven underactuated robotic finger, many parameters need to be satisfied in order to produce a robust robotic hand capable of withstanding industrial environment and capable of fulfilling all needs of robotic assembly in terms of precision and dexterity. For this study, four-bar linkage mechanisms are used to drive underactuated robotic finger and design parameter that will be addressed is transmission performance. Optimization method used for obtaining length of links of four-bar mechanism, based on transmission performance is shown. Freudenstein's analytic method for four-bar linkage function generation, is chosen, and calculated link lengths are to be used for acquiring parameter called transmission defect, parameter that is objective function to be minimized in this optimization process. Maximizing transmission performance, leads to increase of the transmitted torque from the actuated joints to the underactuated joints through transmission mechanism. This paper presents design and kinematic analysis of three degrees of freedom (3-DoF) underactuated robotic finger with linkage driven mechanism for CMSysLab robotic hand.

**Index Terms**—Robotic Assembly, Robotic Hand, Design; Transmission performance, Transmission defect;

## I. INTRODUCTION

Concept of underactuation has many advantages and because of these advantages it is used in many branches of industry. In robotics, it has application in many robotic hands that are designed for laboratory and industrial setting. These hands utilize different mechanisms that transmit the actuation torque to the underactuated joints. Two most popular and widely used concepts for underactuated multifingered robot hands, [1], are tendon and linkage based transmission mechanisms, shown on Fig.1.

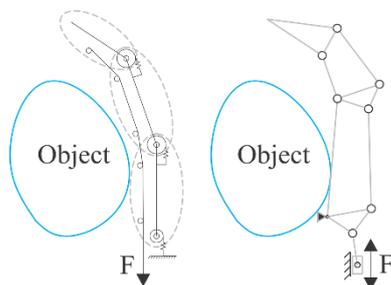


Fig. 1. Tendon and linkage based transmission mechanisms used in underactuated robotic finger. [2].

For our study we have chosen linkage based mechanism because of its rigidity which makes these mechanisms more predictable, more accurate and more controllable.

PhD student Lazar Matijašević is with the Faculty of Mechanical Engineering, University of Belgrade, Kraljice Marije 16, 11120 Belgrade, Serbia (e-mail: [lmatic@mas.bg.ac.rs](mailto:lmatic@mas.bg.ac.rs)).

Full Professor Petar B. Petrović is with the Faculty of Mechanical Engineering, University of Belgrade, Kraljice Marije 16, 11120 Belgrade, Serbia (e-mail: [pbpetrovic@mas.bg.ac.rs](mailto:pbpetrovic@mas.bg.ac.rs)).

These, linkage based mechanisms, are suitable for bigger grasping forces, which is mandatory in assembly operations in industrial setting.

Simplified sketch of underactuated 3-DOF finger with linkage driven mechanism is shown on Fig. 2.

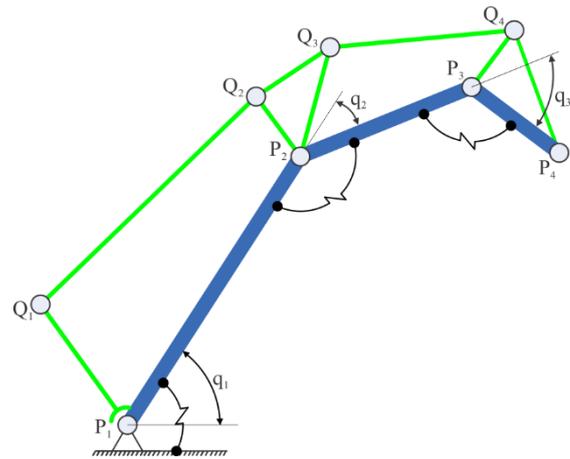


Fig. 2. Representation of 3-DOF finger with linkage driven mechanism.

Transmission mechanism of 3-DOF underactuated finger shown on Fig. 2 consists of two four-linkage mechanisms,  $O_3-O_2-P_2-P_3$  and  $O_2-O_1-P_1-P_2'$ , interconnected with rigid triangular shaped rocker,  $O_2-P_2-P_2'$ .

Requirements for the design of linkage based robotic finger must be met in order to ensure desired motion. Many different optimization algorithms are devised in order to obtain better characteristics of transmission mechanisms. Some of those characteristics are reduced power consumption, reduced structure errors for different mechanisms, force transmission, weight, size and many more. According to aforementioned characteristics there are many design parameters to be met in order to design robust and dexterous robotic hand that can fulfill all requirements of robotic assembly in industrial setting. Focus of this paper will be on transmission performance optimization. Proposed method of optimization focuses on introducing and minimizing parameter named transmission defect and, based on its minimal value, determining optimal value of lengths of four-bar mechanism links.

## II. GENERAL CONCEPT OF ANALYSIS

Variety of useful mechanisms can be formed from a four-link mechanism, shown on Fig. 3, through slight variations, such as changing the character of the pairs, proportions of links, etc. In this paper, two four-bar linkage mechanisms [3], connected with rocker are used for movement and force transmission on phalanx of the underactuated robotic finger.

### A. Freudenstein's equation

Analytical method of kinematic synthesis for four-bar linkage mechanism used in this paper is Freudenstein's

method [4]. Using this method, it is possible to calculate link lengths that accommodate generated function. Also this method is useful in calculating link lengths of four-bar linkage mechanisms because we can use only three positions of mechanism to solve Freudenstein's equation by solving system of three linear equations as will be shown. On Fig 3., a typical four-bar linkage mechanism is shown.

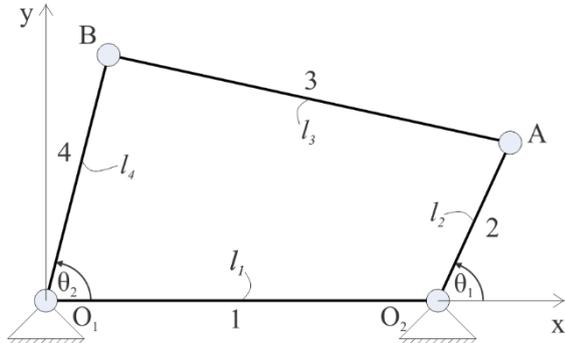


Fig. 3. Representation of four-bar linkage mechanism.

Following parameters are presented on Fig. 3 of  $O_1O_2AB$  four-bar linkage mechanism:

- Fixed link  $O_1O_2$ , ( $l_1$ ), also a phalange of robotic finger,
- Input link  $O_2A$ , ( $l_2$ ),
- Coupler link  $AB$ , ( $l_3$ ),
- Output link  $BO_1$ , ( $l_4$ ),
- Orientation of the input and output link is defined by the angles they construct with x axis and those are input angle  $\theta_1$  and output angle  $\theta_2$ .

Using three position synthesis method allows us to solve equation by generating two pairs of prescribed coordinated movement, shown in (1), of the input and output link.

$$(\theta_1^{12}, \theta_2^{12}), (\theta_1^{23}, \theta_2^{23}). \quad (1)$$

From position 1 to position 2 change in the angle  $\theta_i$  is given by  $\theta_i^{12}$  and corresponding change in the angle  $\theta_2$  is given by  $\theta_2^{12}$ . Same stands for movement from position 2 to position 3. These are two pairs of prescribed movement.

In three position synthesis using Freudenstein's method, we assume the values of  $\theta_i$  corresponding to the first configuration ( $\theta_i^1$ ) and the value of  $\theta_2$  corresponding to first configuration ( $\theta_2^1$ ). Using inputs from (1) we can write:

$$\begin{aligned} \theta_1^2 &= \theta_1^1 + \theta_1^{12}; \\ \theta_2^2 &= \theta_2^1 + \theta_2^{12}, \end{aligned} \quad (2)$$

and:

$$\begin{aligned} \theta_1^3 &= \theta_1^2 + \theta_1^{23}; \\ \theta_2^3 &= \theta_2^2 + \theta_2^{23}. \end{aligned} \quad (3)$$

Freudenstein's equation, which is displacement equation for this planar mechanism, must be valid for all three sets of aforementioned input and output angles. Coordinates of points A and B are:

$$\begin{aligned} x_A &= l_1 + l_2 \cos \theta_1; & x_B &= l_4 \cos \theta_2; \\ y_A &= l_2 \sin \theta_1, & y_B &= l_4 \sin \theta_2. \end{aligned} \quad (4)$$

Length of the coupler link 3 is:

$$l_3^2 = (x_A - x_B)^2 + (y_A - y_B)^2. \quad (5)$$

Substitution of (4) in (5) yields:

$$\cos(\theta_1 - \theta_2) = \frac{l_1}{l_4} \cos \theta_1 - \frac{l_1}{l_2} \cos \theta_2 + \frac{l_1^2 + l_2^2 - l_3^2 + l_4^2}{2l_2l_4}. \quad (6)$$

We now introduce term design parameter  $D_i$ ,  $i=1,2,3$ , described as follows:

$$D_1 = \frac{l_1}{l_4}; D_2 = \frac{l_1}{l_2}; D_3 = \frac{l_1^2 + l_2^2 - l_3^2 + l_4^2}{2l_2l_4}, \quad (7)$$

and substitute it in (6), which in return leads to Freudenstein's equation for four-bar linkage mechanism shown on Fig. 3:

$$\cos(\theta_1 - \theta_2) = D_1 \cos \theta_1 - D_2 \cos \theta_2 + D_3. \quad (8)$$

General form of Freudenstein's equation is:

$$\cos(\theta_1^i - \theta_2^i) = D_1 \cos \theta_1^i - D_2 \cos \theta_2^i + D_3; \quad (9)$$

$$i = 1, 2, 3.$$

From there we can solve three linear equations using angle values determined in (2) and (3) and calculate design parameters  $D_i$ ,  $i=1,2,3$ . These design parameters are used to calculate length values  $l_2$ ,  $l_3$ , and  $l_4$ . Length of link  $l_1$  is chosen for example, based on characteristics of an index human finger or any other assumption. These ratios are important because they allow us to scale robotic finger but to keep same relative movement between various links.

### B. Transmission angle

Now, concept of transmission angle [5], needs to be introduced as well, because that is another parameter, besides aforementioned design parameters  $D_i$ ,  $i=1, 2, 3$ , that are used in optimization method described in this paper. Let's assume that input torque  $T_1$  is acting on link  $O_2A$ . That link is now called input link.

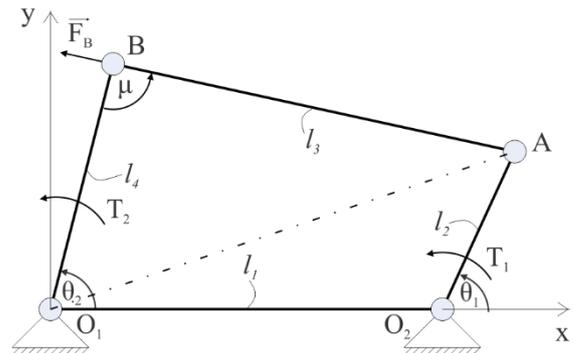


Fig. 4. Representation of transmission angle in four-bar linkage mechanism.

In Fig. 4, if  $O_2A$  is the input link, torque  $T_1$  to the input link is causing force  $\vec{F}_B$ , which is transmitted through the coupler link  $AB$ . For sufficiently slow motions (where we can neglect inertia forces), the force in the coupler link is pure tension or compression and is directed along  $AB$ . For a given force in the coupler link, the torque transmitted to the output link,  $T_2$  (about point  $O_1$ ), is maximum when the angle  $\mu$  between

coupler link  $AB$  and output link  $O_1B$  is  $90^\circ$ . Therefore, angle  $ABO_1$  is called **transmission angle**.

It varies throughout the range of operation from acute angle  $\mu$  to obtuse angle  $\pi - \mu$ .

According to Fig. 4 transmission angle can be calculated by writing the cosine theorem for  $O_1A$  using the triangles  $O_1AB$  and  $O_1AO_2$  and equating the length  $O_1A$ . It is expressed in following form:

$$\mu = \cos^{-1} \frac{l_1^2 + l_2^2 - l_3^2 - l_4^2 - 2l_2l_1 \cos(\pi - \theta_1)}{-2l_3l_4}. \quad (10)$$

Most favorable value of transmission angle is  $90^\circ$  and recommended variations of transmission angle [5] is  $90^\circ \pm 50^\circ$ . When transmission angle is  $0^\circ$  or  $180^\circ$  transmission of motion is impossible. No torque can be realized on output link if transmission angle is  $0^\circ$  and mechanism is at its dead center position.

### C. Transmission quality

The term transmission index [6] was introduced as a generalization of the concept of transmission angle. It is one of many performance indexes of one mechanism. For planar four-bar linkages, the transmission index is shown to be the sine of the transmission angle. Just like the transmission angle, which varies with the configuration of the linkage, the transmission index is also dependent on configuration. As a global performance parameter that measure the force and motion transmission of the linkage, transmission quality [7] is used, which is defined in terms transmission index. Transmission quality is suitable to evaluate mechanism performance throughout its whole range of motion and is not defined only to a local evaluation at a single configuration.

Transmission quality of four-bar linkage mechanism from Fig. 4, is defined in following equation:

$$z = \sqrt{\frac{1}{\theta_1^1 - \theta_1^3} \int_{\theta_1^1}^{\theta_1^3} \sin^2 \mu d\theta_1}. \quad (11)$$

The complement [15] of the transmission quality, is defined as:

$$z' = \sqrt{\frac{1}{\theta_1^1 - \theta_1^3} \int_{\theta_1^1}^{\theta_1^3} \cos^2 \mu d\theta_1}, \quad (12)$$

and is called **transmission defect** which is an objective function to be minimized, and conforms to:

$$z + z' = 1 ; 0 < z' < 1. \quad (13)$$

In this paper, link lengths, which yields minimum transmission defect will be defined. Note that this parameter is only one of many parameters that need to be taken into account when designing underactuated robotic finger, and despite the fact that best transmission of force and movement is desired, lengths of links may differ from values calculated here. It is important to keep in mind that these robotic hands should be as light as possible, robust and in order to be industry acceptable they need to be cost efficient.

It is also stated that with mechanisms that have reversal motion, as robotic hand has (opening and closing of hand)

transmission angle must be investigated for both cases of motion transmission but that will be scope of future research because in this research only pinch and form grips are taken into account, and in those, direction of forces acting upon object of manipulation are only in direction of closing of robotic hand and that direction will be examined.

In next chapter, analysis described above, will be used on optimization of design of CMSysLab underactuated robotic hand.

## III. SYNTHESIS OF OPTIMAL PLANAR FOUR-BAR LINKAGE MECHANISM FOR CMSysLAB UNDERACTUATED ROBOTIC FINGER

When designing transmission mechanism, the crucial issue is selection of mechanism and dimensioning of chosen mechanism. Optimization of planar four-bar linkage [8] mechanism is carried out, as mentioned in previous chapter, with regard to three positions of input and output links which yields three linear equations. These are solved using Freudenstein's equation. Later, optimization of force transmission is carried out using transmission defect as objective function to be minimized. The linkages connected in series are synthesized by starting from the four-bar linkage, which moves the distal phalanx.

### A. Synthesis of the four-bar linkage $O_3, O_2, P_2, P_3$

For synthesis of the function generating four-bar linkage  $O_3, O_2, P_2, P_3$ , shown in Fig. 5, we used method described in section II.

Freudenstein's equation for shown mechanism can be expressed in the form:

$$\cos(\theta_3^i - \theta_4^i) = D_1 \cos \theta_3^i - D_2 \cos \theta_4^i + D_3, \quad i = 1, 2, 3. \quad (14)$$

Design parameters  $D_i, i=1, 2, 3$ , based on (7) are calculated as:

$$D_1 = \frac{L_2}{c_2}; D_2 = \frac{L_2}{a_2}; D_3 = \frac{L_2^2 + a_2^2 - b_2^2 + c_2^2}{2a_2c_2}. \quad (15)$$

The four-bar linkage shown in Fig. 5, consists of links  $a_2, b_2, c_2$  of four-bar linkage:  $O_2P_2, P_2P_3$  and  $P_3O_3$  and medial and distal phalanges  $L_2$  and  $L_3$  of underactuated robotic finger. As described in [1], during motion phalanges move until they come in contact with object that needs to be manipulated. Phalanx that comes in contact with object stops moving, in our case medial phalanx  $L_2$ , which is fixed on Fig. 5, and distal phalanx  $L_3$  can move until it reaches object or mechanical hardstop which constricts its movement.

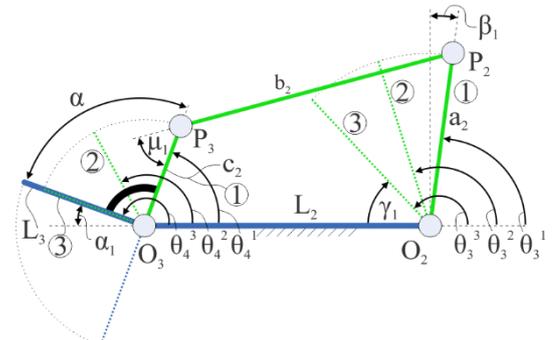


Fig. 5. Representation of four-bar linkage  $O_3, O_2, P_2, P_3$ .

Angles  $\theta_3^i$  and  $\theta_4^i$  for  $i=1, 2, 3$  are input and output angles of mechanism, respectively.

Many of design parameters on Fig. 5 are empirically acquired, based on actual human hand proportions. According to the proposed mechanical design of the finger, the design parameters are  $\alpha = 90^\circ$ ,  $\alpha_1 = 20^\circ$ ,  $\beta_1 = 7.5^\circ$  and  $\gamma_1 = 47^\circ$ . With that in mind, pairs of angles for the starting and final position of links  $a_1$  and  $c_1$  are easily acquired. Values of those angles are  $(\theta_3^1, \theta_4^1) = (82.5^\circ, 70^\circ)$  for starting position and for final position  $(\theta_3^3, \theta_4^3) = (133^\circ, 160^\circ)$ .

Freudenstein's equation (14) can be solved when three positions ①, ② and ③ of the input link  $a_2$ , and corresponding three positions ①, ② and ③ of the output link  $c_2$  are known. Angle pairs, determined above, are to be substituted in (14), and by solving three linear equations we obtain design parameters  $D_i$ ,  $i=1, 2, 3$ , that are used in (15) to calculate link lengths.

Since only two of the three pairs of angles required by the Freudenstein's equations are assigned as design specification of the function generating four-bar linkage  $ABCD$ , an optimization procedure in terms of force transmission has been developed by varying values of  $(\theta_3^2, \theta_4^2)$  during process of optimization.

These values of link lengths are then substituted in transmission angle equation for four-bar linkage mechanism shown on Fig. 5:

$$\mu_1 = \cos^{-1} \frac{L_2^2 + a_2^2 - b_2^2 - c_2^2 - 2L_2 a_2 \cos(\pi - \theta_3)}{2b_2 c_2} \quad (16)$$

Transmission defect for mechanism on Fig. 7 is defined as:

$$z' = \sqrt{\frac{1}{\theta_3^3 - \theta_3^1} \int_{\theta_3^1}^{\theta_3^3} \cos^2 \mu_1 d\theta_3} \quad (17)$$

Substituting (16) in (17) gives the value of  $z'$ . If it is minimum value, calculated lengths of links will be optimal, if not, then angle values  $(\theta_3^2, \theta_4^2)$  should be varied in a specific range until minimum value of  $z'$  is found.

### Proposed algorithm

**Require:** Defining the base phalanx length  $L_2$  and pairs of minimum and maximum values of the input and output angles  $(\theta_3^1, \theta_4^1)$  and  $(\theta_3^3, \theta_4^3)$  of four bar linkage

**Step 1:** Substitute angle pairs into Freudenstein's equation

**Step 2: FOR** (defined increment)

Symbolic solving the system of three equations, resulting in equations for lengths  $a_2$ ,  $b_2$  and  $c_2$ , which are function of angles  $(\theta_3^2, \theta_4^2)$ ;

**end FOR**

**Step 3: FOR** (defined increment)

$(\theta_3^1, \theta_4^1) < (\theta_3^2, \theta_4^2) < (\theta_3^3, \theta_4^3)$

Solve:

$\cos(\theta_3^i - \theta_4^i) = D_1 \cos \theta_3^i - D_2 \cos \theta_4^i + D_3, i = 1, 2, 3.$

Calculated values store in predefined matrices A, B and C

Solve:

$$z' = \sqrt{\frac{1}{\theta_3^3 - \theta_3^1} \int_{\theta_3^1}^{\theta_3^3} \cos^2 \mu_1 d\theta_3}$$

Calculated values store in predefined matrix D

**end FOR**

**Step 4: WHILE**  $z' \sim \min(z')$

Go through matrices A, B, C and D

**end WHILE**

**Step 5:** Obtain values of  $a_2$ ,  $b_2$  and  $c_2$  for  $z' = \min(z')$

Values of link lengths obtained from optimization process are shown in Table III and Fig. 6.

TABLE I  
INPUT PARAMETERS FOR OPTIMIZATION ALGORITHM USED FOR  
OPTIMIZATION OF THE FOUR-BAR LINKAGE O<sub>3</sub>, O<sub>2</sub>, P<sub>2</sub>, P<sub>3</sub>

Minimum and maximum values of input and output angles			
$\theta_3^1$	82.5°	$\theta_3^3$	133°
$\theta_4^1$	70°	$\theta_4^3$	160°

TABLE II  
STARTING VALUES OF PARAMETERS FOR OPTIMIZATION ALGORITHM  
USED FOR OPTIMIZATION OF THE FOUR-BAR LINKAGE O<sub>3</sub>, O<sub>2</sub>, P<sub>2</sub>, P<sub>3</sub>

Starting values of parameters to be optimized with limits of values	
$\theta_3^2$	107.75° ± 25°
$\theta_4^2$	115° ± 45°
$L_2$	37.5 (fixed value)
$a_2$	23mm ± 10mm
$b_2$	36mm ± 10mm
$c_2$	14mm ± 5mm
$z'$	0.103

TABLE III  
OBTAINED PARAMETERS FROM OPTIMIZATION

$z' = 0.02526$			
$L_2$	37.5	$a_2$	28.361
$b_2$	36.993	$c_2$	15.82

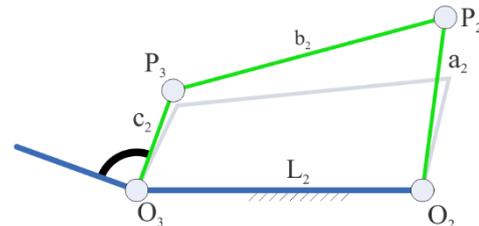


Fig. 6. Optimized parameters of four-bar linkage O<sub>3</sub>, O<sub>2</sub>, P<sub>2</sub>, P<sub>3</sub>.

On Fig. 6, four-bar mechanism of grey color is proportional to starting dimensions of link lengths. Green four-bar mechanism is shown with calculated link lengths. It is obvious that there is difference between calculated values after optimization and starting values. It is because starting values are empirically acquired based on human finger proportions and many more parameters like space limitations, etc. All those parameters need to be taken into account when designing underactuated robotic finger and not just transmission quality as described in this paper. It is obvious that this is iterative process and it needs to accommodate many criteria in order to produce optimal result.

Now dimensions of links of second four-bar linkage will be addressed.

### B. Synthesis of the four-bar linkage $O_2, O_1, P_1, P_2'$

After optimization of four-bar transmission mechanism of medial phalanx it is important to do the same with four-bar transmission mechanism of proximal phalanx.

For synthesis of the function generating four-bar linkage  $O_2, O_1, P_1, P_2'$ , shown in Fig. 7, we used same method as for aforementioned four-bar linkage. Freudenstein's equation for shown mechanism can be expressed in the form:

$$\cos(\theta_1^i - \theta_2^i) = D_1 \cos \theta_1^i - D_2 \cos \theta_2^i + D_3, \quad (18)$$

$$i = 1, 2, 3.$$

Design parameters  $D_i, i=1, 2, 3$ , based on (7) are calculated as:

$$D_1 = \frac{L_1}{c_1}; D_2 = \frac{L_1}{a_1}; D_3 = \frac{L_1^2 + a_1^2 - b_1^2 + c_1^2}{2a_1c_1}. \quad (19)$$

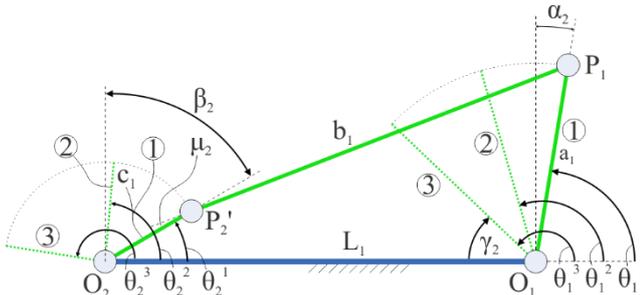


Fig. 7. Representation of four-bar linkage  $O_2, O_1, P_1, P_2'$ .

Parameters shown on Fig. 7:  $L_1$  is the length of proximal phalanx of the finger,  $a_1, b_1$  and  $c_1$  are lengths of links of four-bar linkage:  $O_1P_1, P_1P_2'$  and  $P_2'O_2$ , angles  $\theta_1^i$  and  $\theta_2^i$  for  $i=1, 2, 3$  are input and output angles of mechanism, respectively.

As already shown, (18) can be solved when three positions (1), (2) and (3) of the input link  $a_1$ , and corresponding three positions (1), (2) and (3) of the output link  $c_1$  are predefined.

Many of design parameters on Fig. 7 are empirically acquired, based on actual human hand proportions. According to the proposed mechanical design of the finger, the design parameters are  $\alpha_2 = 10^\circ, \beta_2 = 60^\circ$  and  $\gamma_2 = 43^\circ$ . With that in mind, pairs of angles for the starting and final position of links  $a_1$  and  $c_1$  are easily acquired. Values of those angles are  $(\theta_1^1, \theta_2^1) = (80^\circ, 30^\circ)$  for starting position and for final position  $(\theta_1^3, \theta_2^3) = (137^\circ, 171^\circ)$ . These are to be substituted in (18), and by solving three linear equations we obtain design parameters  $D_i, i=1, 2, 3$ , that are used in (19) to calculate link lengths. These values are then substituted in transmission angle equation:

$$\mu_2 = \cos^{-1} \frac{L_1^2 + a_1^2 - b_1^2 - c_1^2 - 2L_1a_1 \cos(\pi - \theta_1)}{2b_1c_1}. \quad (20)$$

Transmission defect for mechanism on Fig. 7 is defined as

$$z' = \sqrt{\frac{1}{\theta_1^3 - \theta_1^1} \int_{\theta_1^1}^{\theta_1^3} \cos^2 \mu_2 d\theta_1}. \quad (21)$$

Substituting (20) in (21) gives the value of  $z'$ . If it is minimum value, calculated lengths of links will be optimal, if not, then angle values  $(\theta_1, \theta_2)$  should be varied in a specific range until minimum value of  $z'$  is found, the same way as with previous four-bar linkage. Values of link lengths

obtained from optimization process are shown in Table VI and Fig. 8.

TABLE IV  
INPUT PARAMETERS FOR OPTIMIZATION ALGORITHM USED FOR OPTIMIZATION OF THE FOUR-BAR LINKAGE  $O_2, O_1, P_1, P_2'$

Minimum and maximum values of input and output angles			
$\theta_1^1$	80	$\theta_1^3$	137°
$\theta_2^1$	30°	$\theta_2^3$	171°

TABLE V  
STARTING VALUES OF PARAMETERS FOR OPTIMIZATION ALGORITHM USED FOR OPTIMIZATION OF THE FOUR-BAR LINKAGE  $O_2, O_1, P_1, P_2'$

Starting values of parameters to be optimized with limits of values	
$\theta_1^2$	108.5° ± 28°
$\theta_2^2$	100.5° ± 70.5°
$L_1$	37.5 (fixed value)
$a_1$	32mm ± 10mm
$b_1$	60mm ± 10mm
$c_1$	15mm ± 5mm
$z'$	0.081

TABLE VI  
OBTAINED PARAMETERS FROM OPTIMIZATION

$z'=0.02993$			
$L_1$	64.5	$a_1$	41.83
$b_1$	60.554	$c_1$	16.39

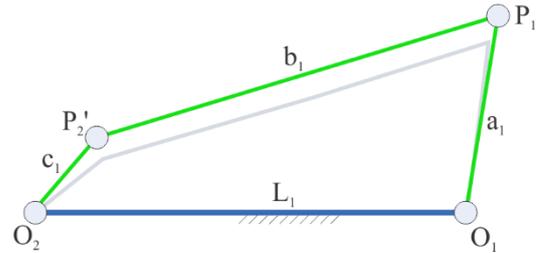


Fig. 8. Optimized parameters of four-bar linkage  $O_2, O_1, P_1, P_2'$ .

On Fig. 8, with grey color, four-bar mechanism with starting link lengths is shown. Green color represents new four-bar mechanism with calculated link lengths.

Previously described method of optimization of four-bar linkage geometry is useful for process of designing linkage based underactuated robotic hand, because calculated values, now can be used as starting values for later designing processes which will use another set of criteria for acquiring optimal geometry of underactuated finger based on many criteria and not just one described in this paper.

## IV. CONCLUSION

In this paper, a method for obtaining optimal lengths of links based on transmission, quality in a four-bar linkage mechanism, was presented. This kind of analysis is important so that researchers and designers of underactuated robotic hands can design proper underactuated finger that can transmit sufficient force onto object that been manipulated with.

As shown, this is only one of many parameters that need to be addressed in order to design proper, robust and dexterous enough robotic hand that is able to accommodate all requirements placed upon it by industrial assembly operations. Some of those requirements are:

**Precision:** assembly process, especially when we take into consideration precise assemblies in auto or aerospace sectors of manufacturing engineering, require high precision and

accuracy of placing of different parts in a assembly.

**Strength of grip:** it is needed in order for robotic hand to be able to counteract contact forces when handling objects. Rigidity and errors that are imposed from lack of rigidity is important factor to take into account. Errors and lack of rigidity are direct consequence of clearances that exist in assemblies.

**Access to the zone of assembly:** access to the zone in which assembly of parts needs to take place in required position and orientation.

#### ACKNOWLEDGMENT

This research work is supported by the Serbian Ministry for Education, Science and Technology Development through the projects titled 'Smart Robotics for Customized Manufacturing', grant No.: TR35007, and 'Development and Experiments of Mobile Collaborative Robot With Dual-Arm', grant No.: 401-00-00589/2018-09, which is jointly realized by Anhui University of Technology, Ma'anshan, Anhui, and Tsinghua University, Beijing, from one side, and Institute Mihajlo Pupin – Belgrade (IMP), Faculty of Mechanical Engineering, University of Belgrade (MFB), and Faculty of technical sciences of Novi Sad, University of Novi Sad (FTN), from the other side, supported by a cluster of

industrial partners from both sides, as a bilateral scientific project of high national relevance with the People's Republic of China.

#### REFERENCES

- [1] Birglen, L., et al. “*Underactuated Robotic Hands*”, Springer, Berlin, 2008.
- [2] Matijasevic, L, Milivojevic, M, Petrovic, P, “Multifingered under-actuated hands in robotic assembly”, *13th International Scientific Conference MMA 2018 Flexible Technologies*, Proceedings, ISBN 978-86-6022-094-5, pp.91-94.
- [3] L. Birglen ; C.M. Gosselin “Kinetostatic Analysis of Underactuated Fingers”, *IEEE Transactions on Robotics and Automation*, vol. 20 Issue: 1, pp.211 -221, april 2004
- [4] F. Freudenstein, “Approximate Synthesis of Four-Bar Linkages”, *ASME Trans.*, Vol.77, No.8, pp.853–861, August 1955.
- [5] S. S. Balli ; S. Chand “Transmission angle in mechanisms (Triangle in mech)”, in *Mechanism and Machine Theory*, Elsevier vol. 37 Issue: 2, pp.175-195, 2002.
- [6] Chen, C., & Angeles, J.: “Generalized transmission index and transmission quality for spatial linkages”. *Mechanism and Machine Theory*, 42(9), 1225–1237, 2007.
- [7] Angeles, J., & Bernier, A.: “The Global Least-Square Optimization of Function-Generating Linkages”, *Journal of Mechanisms Transmissions and Automation in Design*, 109(2), 204, 1987.
- [8] C. Gosselin ; J. Angeles “Optimization of planar and spherical function generators as minimum-defect linkages”, in *Mechanism and Machine Theory*, Elsevier vol. 24 Issue: 4, pp.293-307, 1989.

# Calculation of achievable robot joint accelerations based on a new robot forward dynamics algorithm

Vladimir Kvrđic, Jelena Vidakovic

**Abstract**— An algorithm that calculates the feasible robot joints' accelerations based on a new forward dynamics algorithm while considering the actuators' force/torque saturations and achieves a realistic simulation of robot movements is given in this paper. While the most used forward dynamics algorithm in the literature, Walker and Orin's Method 1, calculates robot forward dynamics by executing Recursive Newton-Euler Algorithm (RNEA)  $n + 1$  times, where  $n$  is the number of degrees-of-freedom (DoFs), algorithm used here solves forward dynamics using the modified RNEA (mRNEA) only once. Owing to that, this algorithm is very efficient. Furthermore, the computational complexity of the algorithm is even more significant when used for robot simulation as it does not require calculating joint torques as inputs for forward dynamics, unlike other methods. Another benefit of the proposed method is the ease of development and implementation for a specific robot. The proposed mRNEA and its application within the forward dynamics algorithm are demonstrated using a serial 4-DoF spatial disorientation trainer as an example.

**Index Terms**—Robot, Forward dynamics, Joint accelerations, Simulation system, Recursive Newton–Euler algorithm

## I. INTRODUCTION

A robot simulation verifies the feasibility of programmed movements, and if necessary modifies them. It also calculates the values of forces and moments acting on robot links and joints that is essential in robot design. For this, robot simulation has to solve robot forward and inverse dynamics problems. Forward dynamics (FD) solves the motion from the forces, while inverse dynamics (ID) solves the forces from the motion [1]. ID is used within dynamic model-based control methods, and for FD calculations. FD is used mainly in simulation purposes.

FD calculates the joint accelerations  $\ddot{\mathbf{q}}(t_k)$  at a time instant  $t_k$ , the joint velocities  $\dot{\mathbf{q}}(t_{k+1})$  of the next interpolation cycle time ( $\Delta t$ ) and joint positions  $\mathbf{q}(t_{k+1})$  at the end of the next  $\Delta t$ . FD accounts for the joint torques  $\mathbf{u}(t_k)$ ; the inertial, gravitational, and Coriolis forces of the robot links; forces and moments acting on the end effector; and the friction forces and moments of the joints.

When a robot is considered as a continuous nonlinear system, after obtaining  $\ddot{\mathbf{q}}(t_k)$ , the velocity  $\dot{\mathbf{q}}(t_{k+1})$  and position  $\mathbf{q}(t_{k+1})$ ,  $t_{k+1} = t_k + \Delta t$  are computed using a numerical integration method, i.e., Runge–Kutta, with an integration step  $\Delta t$  [2]. On

the other hand, ID determines the joint torques  $\mathbf{u}(t_k)$  at time instant  $t_k$  which are required to generate the motion specified by the joint accelerations, and consequently, the velocities and positions. This is accomplished by using the current velocities, current positions, the forces and moments acting on the end effector, and the friction forces and moments of the joints.

One of the challenges in robot simulation is to derive algorithms that are computationally efficient and are also easy to apply to a specific robot.

The ID of a manipulator with  $n$  DoFs can be solved by well-known equations of motion which represent its joint space dynamic model

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + \mathbf{J}(\mathbf{q})^T \mathbf{k}_e = \mathbf{u}, \quad (1)$$

where  $\ddot{\mathbf{q}}$ ,  $\dot{\mathbf{q}}$ , and  $\mathbf{q}$  are  $n \times 1$  vectors of the joint accelerations, velocities, and positions, respectively;  $\mathbf{H}(\mathbf{q})$  is  $n \times n$  generalized robot mass (inertia) matrix;  $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$  is an  $n \times n$  matrix specifying the centrifugal and Coriolis effects;  $\mathbf{g}(\mathbf{q})$  is an  $n \times 1$  vector of gravity terms;  $\mathbf{k}_e$  is a  $6 \times 1$  vector of the external forces and moments on link  $n$ ;  $\mathbf{J}(\mathbf{q})$  is a  $6 \times n$  Jacobian matrix; and  $\mathbf{u}$  is an  $n \times 1$  vector of the input joint torques/forces. The diagonal terms of the mass matrix are related to the inertias of the corresponding DoF, and the off-diagonal terms express the inertial couplings between the DoFs [3].

From Eq. (1), it can be seen that for time instant  $t_k$ , the joint torques/forces are linear functions of the joint accelerations  $\ddot{\mathbf{q}}(t_k)$  when  $\mathbf{q}(t_k)$  and  $\dot{\mathbf{q}}(t_k)$  are given. These equations can be obtained explicitly with the Lagrange formulation (LF) which contains the matrix  $\mathbf{H}(\mathbf{q})$  and vectors  $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}$ ,  $\mathbf{g}(\mathbf{q})$ , and  $\mathbf{J}(\mathbf{q})^T \mathbf{k}_e$ . Consequently, the joint accelerations  $\ddot{\mathbf{q}}(t_k)$  can be computed by solving the following system of  $n$  linear equations

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} = \mathbf{u} - \mathbf{u}'(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{k}_e), \quad (2)$$

$$\mathbf{u}'(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{k}_e) = \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + \mathbf{J}(\mathbf{q})^T \mathbf{k}_e. \quad (3)$$

The LF method for derivation of robot equations of motion provides a compact analytical form containing the mass matrix  $\mathbf{H}(\mathbf{q})$ , and a bias vector  $\mathbf{u}'$  that denotes joint torque contributions that do not correlate with the joint accelerations

Vladimir Kvrđic is with the Institute Mihajlo Pupin, University of Belgrade, 15 Volgina, 11060 Belgrade, Serbia, [vladimir.kvrđic@pupin.rs](mailto:vladimir.kvrđic@pupin.rs).

Jelena Vidakovic is with the Lola Institute, Kneza Visislava 70a, 11030 Belgrade, Serbia, [jelena.vidakovic@li.rs](mailto:jelena.vidakovic@li.rs).

[4–10]. Therefore, when the LF is used, the joint accelerations  $\ddot{\mathbf{q}}(t_k)$  (within one interpolation cycle) can be computed by solving the system of  $n$  linear equations, where  $n$  is number of manipulator DoFs. Although it is not complex to solve FD using LF, this method is typically used for manipulators when  $n \leq 3$  because of the very high computational complexity of the LF:  $O(n^4)$ .

Robot dynamic models should be derived in recursive form in order to be computational efficient [5]. In contrast to the LF, the computational complexity of the RNEA is  $O(n)$ . Reference [11] developed a recursive LF; however, the computational complexity of the recursive LF is  $O(n^3)$ .

Walker and Orin [6,7] employed the RNEA for computing the FD and presented four methods to solve the joint accelerations. Their method 1 (WO method 1) remains the simplest and the most recommended in the literature [2,5]. According to this method, torque  $\mathbf{u}'$  is computed using the RNEA. Further, each column  $\mathbf{h}_i$ ,  $i = 1$  to  $n$ , of matrix  $\mathbf{H}$  is computed as the torque vector given by the RNEA.

In [12], a modified Recursive Newton-Euler Algorithm (mRNEA) for derivation of dynamic model of robot manipulator is presented. The mRNEA gives explicitly the mass matrix  $\mathbf{H}$  and the bias vector  $\mathbf{u}'$ , in a similar manner as LF. Owing to that, it is easy to use in the FD computation, which executes the mRNEA only once.

The additional calculation of the input-joint-torque  $\mathbf{u}$ , Eqs. (2), had to be performed within a robot simulation system. With the method presented in [12], input-joint-torque calculation is performed within the FD algorithm, and therefore, the computational complexity of the simulation system is additionally reduced. As a result, the simulation system has to solve ID only once within each interpolation cycle, in comparison to simulation systems which use, for example, WO method 1, and which solve ID  $n + 2$  times.

Proposed FD algorithm is computationally very efficient, with  $O(n)$  complexity.

The rest of the paper is organized as follows. Section II presents the proposed approach for efficient FD calculation and its implementation in a robot simulation system. Section III depicts the proposed FD algorithm for open-chain manipulators with  $n$  DoFs. A toy model for the 4-DoF spatial disorientation trainer (SDT) is also presented in Section IV. The FD algorithm, which calculates the achievable motor velocities in each interpolation cycle based on the actuator torque/force saturations, is presented in Section V. Finally, concluding remarks are given in Section VI.

## II. ACHIEVABLE JOINT ACCELERATIONS CALCULATION

A path planner of the robot controller transforms the motion commands into a series of successive positions of robot joints/actuators. As they are sent to the servo controller at constant time intervals  $\Delta t$ , they correspond to the desired joint/actuator velocities of  $\dot{q}_i(t_k) = (q_i(t_{k+1}) - q_i(t_k)) / \Delta t$ , which can be considered constant within each  $\Delta t$  (up-to-date controllers have  $\Delta t$  between 0.01 s and 0.003 s). Thereafter, the path planner sends the desired joint velocities to the speed controllers of the actuators, whose task is to keep them constant

within each  $\Delta t$ .

Since each joint velocity can be considered constant within each  $\Delta t$ , and since the current velocity  $\dot{q}_i(t_k)$  and given acceleration  $\ddot{q}_i(t_k)$  are known (calculated within the path planner), the joint velocity in the next interpolation cycle is  $\dot{q}_i(t_{k+1}) = \dot{q}_i(t_k) + \ddot{q}_i(t_k) \Delta t$ , which is depicted in Fig. 1.

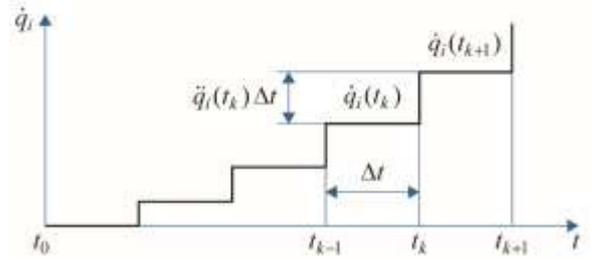


Fig. 1. Example of given joint velocity change.

Herein, a method for obtaining equation (2) explicitly with the mRNEA is proposed. Consequently, a method which calculates  $\ddot{\mathbf{q}}(t_k)$  using the mRNEA only once is proposed. The presented FD algorithm utilizes the current values of  $\mathbf{q}(t_k)$  and  $\dot{\mathbf{q}}(t_k)$ , the given values of  $\mathbf{q}(t_{k+1})$  and  $\dot{\mathbf{q}}(t_{k+1})$  calculated in the path interpolator, and  $\mathbf{k}_e$ . First, it checks if the desired positions and velocities are feasible. If they are not, it limits their values in accordance with their maximum/minimum possible values. Based on this, the algorithm calculates the desired joint accelerations  $\ddot{q}_i(t_k) = (\dot{q}_i(t_{k+1}) - \dot{q}_i(t_k)) / \Delta t$ . Next, the mRNEA calculates the joint torques/forces  $\mathbf{u}$  required for the desired joint motions. In the next step, the algorithm calculates the required actuator torques  $\mathbf{u}_a$ , whose capabilities are examined. Unachievable torques/forces are replaced with the maximum/minimum possible, with the aim that the FD algorithm determines the achievable accelerations. Other joint accelerations keep their values obtained from the path interpolator.

Herein, within the simulation system, only attainable motor velocities and positions are sent from the path planner to the speed controller during each  $\Delta t$ . Consequently, joint forces and moments are calculated based on the attainable velocities and accelerations, so that their realistic values are obtained.

The FD simulation can be used in a stage of the robot design process, in which case it enables the proper design of bearings and links.

## III. FD ALGORITHM BASED ON MRNEA

Herein, the FD algorithm based on mRNEA for open-chain manipulators with  $n$  DoFs is presented.

A  $4 \times 4$  homogenous transformation matrix (HTM) that transforms point coordinates from frame  $j$  to frame  $i$  is  ${}^j\mathbf{T}_i$ , and from the base frame to frame  $i$  is  $\mathbf{T}_i$ . The matrix  ${}^j\mathbf{T}_i$  contains a  $3 \times 3$  orientation matrix  ${}^j\mathbf{D}_i = [{}^j\mathbf{x}_i \quad {}^j\mathbf{y}_i \quad {}^j\mathbf{z}_i]$  and a  $3 \times 1$  position vector  ${}^j\mathbf{p}_i$ .

The linear acceleration of the robot link  $i$  centre of mass is

$$\dot{\mathbf{v}}_i^{\text{cm}} = [\dot{v}_{xi}^{\text{cm}} \ \dot{v}_{yi}^{\text{cm}} \ \dot{v}_{zi}^{\text{cm}}]^T = \dot{\mathbf{v}}_i + \dot{\boldsymbol{\omega}}_i \times \mathbf{r}_i^{\text{cm}} + \boldsymbol{\omega}_i \times (\boldsymbol{\omega}_i \times \mathbf{r}_i^{\text{cm}}), \quad (4)$$

where  $\mathbf{r}_i^{\text{cm}} = [r_{xi}^{\text{cm}} \ r_{yi}^{\text{cm}} \ r_{zi}^{\text{cm}}]^T = [r_{xi} \ r_{yi} \ r_{zi}]^T = \mathbf{D}_i \hat{\mathbf{r}}_i^{\text{cm}}$  is the position of the link  $i$  centre of mass with respect to the coordinates of link  $i$  expressed in the base coordinates. This vector in the coordinates of link  $i$  is  $\hat{\mathbf{r}}_i^{\text{cm}} = [\hat{r}_{xi} \ \hat{r}_{yi} \ \hat{r}_{zi}]^T$ . A vector cross product is denoted with  $\times$ , and  $\boldsymbol{\omega}_i$ ,  $\dot{\boldsymbol{\omega}}_i$ , and  $\dot{\mathbf{v}}_i$  are the link angular velocity, angular acceleration, and linear acceleration, respectively,  $i = 1$  to  $n$ . Equation (4) can be rewritten as

$$\dot{\mathbf{v}}_i^{\text{cm}} = \sum_{k=1}^i \mathbf{b}_{ik} \ddot{q}_k + \sum_{k=1}^i \sum_{j=k}^i \mathbf{b}_{ikj} \dot{q}_k \dot{q}_j, \quad i = 1 \text{ to } n, \quad (5)$$

where  $\mathbf{b}_{ik}$  and  $\mathbf{b}_{ikj}$  are  $3 \times 1$  vectors. The total force  $\mathbf{F}_i$  and total moment  $\mathbf{N}_i$  exerted on link  $i$ , obtained from the NE equations, are

$$\mathbf{F}_i = [F_{xi} \ F_{yi} \ F_{zi}]^T = m_i [\dot{v}_{xi}^{\text{cm}} \ \dot{v}_{yi}^{\text{cm}} \ \dot{v}_{zi}^{\text{cm}} - g]^T, \quad (6)$$

$$\mathbf{N}_i = [N_{xi} \ N_{yi} \ N_{zi}]^T = \mathbf{I}_i^{\text{cm}} \dot{\boldsymbol{\omega}}_i + \boldsymbol{\omega}_i \times (\mathbf{I}_i^{\text{cm}} \boldsymbol{\omega}_i). \quad (7)$$

The mass of link  $i$  is denoted as  $m_i$ ,  $g$  is Earth's acceleration and  $\mathbf{I}_i^{\text{cm}}$  is the  $3 \times 3$  moment of the inertia matrix of link  $i$  about the centre of mass of that link expressed in the base coordinates. Equations (5) and (6) yield

$$\mathbf{F}_i = m_i ([0 \ 0 \ -g]^T + \sum_{k=1}^i \mathbf{b}_{ik} \ddot{q}_k + \sum_{k=1}^i \sum_{j=k}^i \mathbf{b}_{ikj} \dot{q}_k \dot{q}_j), \quad i = 1 \text{ to } n. \quad (8)$$

Equation (7) can be rewritten as

$$\mathbf{N}_i = \sum_{k=1}^i \mathbf{d}_{ik} \ddot{q}_k + \sum_{k=1}^i \sum_{j=k}^i \mathbf{d}_{ikj} \dot{q}_k \dot{q}_j, \quad i = 1 \text{ to } n, \quad (9)$$

where  $\mathbf{d}_{ik}$  and  $\mathbf{d}_{ikj}$  are  $3 \times 1$  vectors. The effects of the external forces and moments,  $\mathbf{k}_e^T = [\mathbf{f}_e \ \mathbf{n}_e]^T$ , acting on the end effector are well-known as

$$\mathbf{f}_n = \mathbf{F}_n + \mathbf{f}_e, \quad (10)$$

$$\mathbf{n}_n = \mathbf{N}_n + \mathbf{n}_e + [p_{ey} f_{ez} - p_{ez} f_{ey} \ p_{ez} f_{ex} - p_{ex} f_{ez} \ p_{ex} f_{ey} - p_{ey} f_{ex}]^T, \quad (11)$$

where  $\mathbf{p}_e = [p_{ex} \ p_{ey} \ p_{ez}]^T = \mathbf{D}_n \hat{\mathbf{p}}_e$  is the position of the external force with respect to the coordinates of link  $n$  expressed in the base coordinates. This vector in the coordinates of link  $n$  is  $\hat{\mathbf{p}}_e = [\hat{p}_{ex} \ \hat{p}_{ey} \ \hat{p}_{ez}]^T$ . Equations (8) and (10) yield

$$\mathbf{F}_i = \mathbf{e}_i + \sum_{k=1}^i \mathbf{e}_{ik} \ddot{q}_k, \quad (12)$$

$$\mathbf{e}_i = m_i ([0 \ 0 \ -g]^T + \sum_{k=1}^i \sum_{j=k}^i \mathbf{b}_{ikj} \dot{q}_k \dot{q}_j), \quad i = 1 \text{ to } n-1, \quad (13)$$

$$\mathbf{e}_n = m_n ([0 \ 0 \ -g]^T + \sum_{k=1}^n \sum_{j=k}^n \mathbf{b}_{ikj} \dot{q}_k \dot{q}_j) + \mathbf{f}_e, \quad i = n, \quad (14)$$

$$\mathbf{e}_{ik} = m_i \mathbf{b}_{ik}, \quad i = 1 \text{ to } n. \quad (15)$$

Similarly, Eq. (9) can be replaced with

$$\mathbf{N}_i = \mathbf{d}_i + \sum_{k=1}^i \mathbf{d}_{ik} \ddot{q}_k, \quad (16)$$

$$\mathbf{d}_i = \sum_{k=1}^i \sum_{j=k}^i \mathbf{d}_{ikj} \dot{q}_k \dot{q}_j, \quad i = 1 \text{ to } n. \quad (17)$$

From robot dynamics, the force  $\mathbf{f}_i$  and moment  $\mathbf{n}_i$  exerted on link  $i$  by link  $i-1$  in the base coordinate frame, is well-known to be

$$\mathbf{f}_i = [f_{xi} \ f_{yi} \ f_{zi}]^T = \mathbf{F}_i + \mathbf{f}_{i+1}, \quad (18)$$

$$\mathbf{n}_i = [n_{xi} \ n_{yi} \ n_{zi}]^T = \mathbf{n}_{i+1} + \mathbf{N}_i + \mathbf{l}_i \times \mathbf{F}_i + \mathbf{p}_i^* \times \mathbf{f}_{i+1}, \quad (19)$$

where  $\mathbf{l}_i = [l_{xi} \ l_{yi} \ l_{zi}]^T = \mathbf{p}_i^* + \mathbf{r}_i^{\text{cm}}$ ,  $\mathbf{p}_i^* = \mathbf{p}_{i+1} - \mathbf{p}_i$ .

The mass of the end effector can be included in the mass of link  $n$ . In accordance with Eqs. (10)–(19),  $\mathbf{f}_i$  and  $\mathbf{n}_i$  can be calculated as

$$\mathbf{f}_i = \sum_{k=i}^n \mathbf{e}_k + \sum_{k=1}^i \sum_{j=i}^n \mathbf{e}_{jk} \ddot{q}_k + \sum_{k=i+1}^n \sum_{j=k}^n \mathbf{e}_{jk} \ddot{q}_k, \quad i = n \text{ to } 1, \quad (20)$$

$$\begin{aligned} \mathbf{n}_i &= \mathbf{n}_{i+1} + \mathbf{d}_i + \sum_{k=1}^i \mathbf{d}_{ik} \ddot{q}_k + \mathbf{l}_i \times (\mathbf{e}_i + \sum_{k=1}^i \mathbf{e}_{ik} \ddot{q}_k) \\ &+ \mathbf{p}_i^* \times (\sum_{k=i+1}^n \mathbf{e}_k + \sum_{k=1}^{i+1} \sum_{j=i+1}^n \mathbf{e}_{jk} \ddot{q}_k + \sum_{k=i+2}^n \sum_{j=k}^n \mathbf{e}_{jk} \ddot{q}_k) \end{aligned} \quad (21)$$

$$= \mathbf{n}_{i+1} + \mathbf{n}_{ic} + \mathbf{n}_{ia1} + \mathbf{n}_{ia2} + \mathbf{n}_{ia3}, \quad i = n \text{ to } 1,$$

where  $\mathbf{n}_{ic}$ ,  $\mathbf{n}_{ia1}$ ,  $\mathbf{n}_{ia2}$ , and  $\mathbf{n}_{ia3}$  are  $3 \times 1$  vectors, as follows:

$$\mathbf{n}_{nc} = \mathbf{d}_n + \mathbf{l}_n \times \mathbf{e}_n + \mathbf{p}_{n+1} \times \mathbf{f}_{n+1}$$

$$= \begin{bmatrix} d_{xn} + l_{yn} e_{zn} - l_{zn} e_{yn} + p_{y(n+1)} f_{z(n+1)} - p_{z(n+1)} f_{y(n+1)} \\ d_{yn} + l_{zn} e_{xn} - l_{xn} e_{zn} + p_{z(n+1)} f_{x(n+1)} - p_{x(n+1)} f_{z(n+1)} \\ d_{zn} + l_{xn} e_{yn} - l_{yn} e_{xn} + p_{x(n+1)} f_{y(n+1)} - p_{y(n+1)} f_{x(n+1)} \end{bmatrix}, i = n, \quad (22)$$

$$\mathbf{n}_{ic} = \mathbf{d}_i + \mathbf{l}_i \times \mathbf{e}_i + \mathbf{p}_i^* \times \mathbf{E}_{i+1}$$

$$= \begin{bmatrix} d_{xi} + l_{yi} e_{zi} - l_{zi} e_{yi} + p_{yi}^* E_{z(i+1)} - p_{zi}^* E_{y(i+1)} \\ d_{yi} + l_{zi} e_{xi} - l_{xi} e_{zi} + p_{zi}^* E_{x(i+1)} - p_{xi}^* E_{z(i+1)} \\ d_{zi} + l_{xi} e_{yi} - l_{yi} e_{xi} + p_{xi}^* E_{y(i+1)} - p_{yi}^* E_{x(i+1)} \end{bmatrix}, \quad (23)$$

$$\mathbf{E}_{i+1} = \sum_{k=i+1}^n \mathbf{e}_k, \quad i = n-1 \text{ to } 1,$$

$$\mathbf{n}_{ia1} = \sum_{k=1}^i (\mathbf{d}_{ik} + \mathbf{l}_i \times \mathbf{e}_{ik}) \ddot{q}_k = \begin{bmatrix} \sum_{k=1}^i (d_{xik} + l_{yi} e_{zik} - l_{zi} e_{yik}) \ddot{q}_k \\ \sum_{k=1}^i (d_{yik} + l_{zi} e_{xik} - l_{xi} e_{zik}) \ddot{q}_k \\ \sum_{k=1}^i (d_{zik} + l_{xi} e_{yik} - l_{yi} e_{xik}) \ddot{q}_k \end{bmatrix}$$

$$= [\mathbf{n}_{i1a1} \quad \mathbf{n}_{i2a1} \quad \dots \quad \mathbf{n}_{iia1}] [\ddot{q}_1 \quad \ddot{q}_2 \quad \dots \quad \ddot{q}_i]^T,$$

$$[\mathbf{n}_{i1a1} \quad \mathbf{n}_{i2a1} \quad \dots \quad \mathbf{n}_{iia1}] = [\mathbf{d}_{i1} + \mathbf{l}_i \times \mathbf{e}_{i1} \quad \mathbf{d}_{i2} + \mathbf{l}_i \times \mathbf{e}_{i2} \quad \dots \quad \mathbf{d}_{ii} + \mathbf{l}_i \times \mathbf{e}_{ii}], \quad (24)$$

$$i = n \text{ to } 1,$$

$$\mathbf{n}_{ia2} = \mathbf{p}_i^* \times \sum_{k=1}^{i+1} \mathbf{E}_{(i+1)k} \ddot{q}_k = \begin{bmatrix} \sum_{k=1}^{i+1} (p_{yi}^* E_{z(i+1)k} - p_{zi}^* E_{y(i+1)k}) \ddot{q}_k \\ \sum_{k=1}^{i+1} (p_{zi}^* E_{x(i+1)k} - p_{xi}^* E_{z(i+1)k}) \ddot{q}_k \\ \sum_{k=1}^{i+1} (p_{xi}^* E_{y(i+1)k} - p_{yi}^* E_{x(i+1)k}) \ddot{q}_k \end{bmatrix}$$

$$= [\mathbf{n}_{i1a2} \quad \mathbf{n}_{i2a2} \quad \dots \quad \mathbf{n}_{i(i+1)a2}] [\ddot{q}_1 \quad \ddot{q}_2 \quad \dots \quad \ddot{q}_{i+1}]^T,$$

$$[\mathbf{n}_{i1a2} \quad \mathbf{n}_{i2a2} \quad \dots \quad \mathbf{n}_{i(i+1)a2}] = [\mathbf{p}_i^* \times \mathbf{E}_{(i+1)1} \quad \mathbf{p}_i^* \times \mathbf{E}_{(i+1)2} \quad \dots \quad \mathbf{p}_i^* \times \mathbf{E}_{(i+1)(i+1)}],$$

$$\mathbf{E}_{(i+1)k} = \sum_{j=i+1}^n \mathbf{e}_{jk}, \quad i = n-1 \text{ to } 1, \quad \mathbf{n}_{na2} = \mathbf{0}, \quad (25)$$

$$\mathbf{n}_{ia3} = \mathbf{p}_i^* \times \sum_{k=i+2}^n \mathbf{E}_{kk} \ddot{q}_k = \begin{bmatrix} \sum_{k=i+2}^n (p_{yi}^* E_{zkk} - p_{zi}^* E_{ykk}) \ddot{q}_k \\ \sum_{k=i+2}^n (p_{zi}^* E_{xkk} - p_{xi}^* E_{zkk}) \ddot{q}_k \\ \sum_{k=i+2}^n (p_{xi}^* E_{ykk} - p_{yi}^* E_{xkk}) \ddot{q}_k \end{bmatrix}$$

$$= [\mathbf{0} \quad \mathbf{0} \quad \dots \quad \mathbf{n}_{i(i+2)a3} \quad \dots \quad \mathbf{n}_{ina3}] [\ddot{q}_1 \quad \ddot{q}_2 \quad \dots \quad \ddot{q}_{i+2} \quad \dots \quad \ddot{q}_n]^T,$$

$$[\mathbf{0} \quad \mathbf{0} \quad \dots \quad \mathbf{n}_{i(i+2)a3} \quad \dots \quad \mathbf{n}_{ina3}] = [\mathbf{0} \quad \mathbf{0} \quad \dots \quad \mathbf{p}_i^* \times \mathbf{E}_{(i+2)(i+2)} \quad \dots \quad \mathbf{p}_i^* \times \mathbf{E}_{im}],$$

$$\mathbf{E}_{kk} = \sum_{j=k}^n \mathbf{e}_{jk}, \quad i = n-2 \text{ to } 1, \quad \mathbf{n}_{na3} = \mathbf{n}_{(n-1)a3} = \mathbf{0}. \quad (26)$$

In order to reduce the number of counts, the vector  $\mathbf{n}_{i+1}$  can be included in the vectors  $\mathbf{n}_{ic}$ ,  $\mathbf{n}_{ia1}$ ,  $\mathbf{n}_{ia2}$ , and  $\mathbf{n}_{ia3}$ . In this way, Eq. (21) is transformed into

$$\mathbf{n}_i = \mathbf{n}'_{ic} + (\mathbf{n}'_{ia1} + \mathbf{n}'_{ia2} + \mathbf{n}'_{ia3}) \ddot{\mathbf{q}}, \quad i = n \text{ to } 1, \quad (27)$$

where  $\ddot{\mathbf{q}} = [\ddot{q}_1 \quad \ddot{q}_2 \quad \dots \quad \ddot{q}_n]^T$  is an  $n \times 1$  vector,  $\mathbf{n}'_{ic}$  is a  $3 \times 1$  vector, and  $\mathbf{n}'_{ia1}$ ,  $\mathbf{n}'_{ia2}$ , and  $\mathbf{n}'_{ia3}$  are  $3 \times n$  vectors. They are given in the following equations:

$$\mathbf{n}'_{ic} = \mathbf{n}_{(i+1)c} + \mathbf{n}_{ic}, \quad (28)$$

$$\mathbf{n}'_{ia1} = [\mathbf{n}'_{i1a1} \quad \mathbf{n}'_{i2a1} \quad \dots \quad \mathbf{n}'_{ima1}]$$

$$= \left[ \sum_{k=i}^n (\mathbf{d}_{k1} + \mathbf{l}_k \times \mathbf{e}_{k1}) \quad \sum_{k=i}^n (\mathbf{d}_{k2} + \mathbf{l}_k \times \mathbf{e}_{k2}) \quad \dots \quad \mathbf{d}_{nn} + \mathbf{l}_n \times \mathbf{e}_{nn} \right], \quad (29)$$

$$\mathbf{n}'_{ia2} = \left[ \sum_{k=i}^{n-1} \mathbf{n}_{k1a2} \quad \sum_{k=i}^{n-1} \mathbf{n}_{k2a2} \quad \sum_{k=i}^{n-1} \mathbf{n}_{k3a2} \quad \sum_{k=i}^{n-1} \mathbf{n}_{k4a2} \quad \dots \quad \mathbf{n}_{(n-1)(n-1)a2} \right]$$

$$= \left[ \sum_{k=i}^{n-1} (\mathbf{p}_k^* \times \mathbf{E}_{(k+1)1}) \quad \sum_{k=i}^{n-1} (\mathbf{p}_k^* \times \mathbf{E}_{(k+1)2}) \quad \sum_{k=i}^{n-1} (\mathbf{p}_k^* \times \mathbf{E}_{(k+1)3}) \right.$$

$$\left. \sum_{k=i}^{n-1} (\mathbf{p}_k^* \times \mathbf{E}_{(k+1)4}) \quad \dots \quad \mathbf{n}_{(n-1)(n-1)a2} \right], \quad (30)$$

$$\mathbf{n}'_{ia3} = \left[ \mathbf{0} \quad \mathbf{0} \quad \mathbf{p}_i^* \times \mathbf{E}_{33} \quad (\mathbf{p}_i^* + \mathbf{p}_2^*) \times \mathbf{E}_{44} \quad \dots \quad \sum_{i=1}^{n-3} \mathbf{p}_i^* \times \mathbf{E}_{(n-1)(n-1)} \quad \sum_{i=1}^{n-2} \mathbf{p}_i^* \times \mathbf{e}_{nm} \right]. \quad (31)$$

The forces and moments exerted on link  $i$  by link  $i-1$  in the coordinates of link  $i-1$  are

$$\hat{\mathbf{f}}_i = [\hat{f}_{xi} \quad \hat{f}_{yi} \quad \hat{f}_{zi}]^T = \mathbf{D}_{i-1}^T \mathbf{f}_i \quad \text{and} \quad \hat{\mathbf{n}}_i = [\hat{n}_{xi} \quad \hat{n}_{yi} \quad \hat{n}_{zi}]^T = \mathbf{D}_{i-1}^T \mathbf{n}_i. \quad (32)$$

The projection of  $\mathbf{n}_i$  along the axis of motion of joint  $i$  is

$$u_i = \mathbf{z}_{i-1}^T \mathbf{n}_i, \quad (33)$$

where  $\mathbf{z}_{i-1}$  is a unit vector of the axis of motion, given in the first three elements of the third column of the matrix  $\mathbf{T}_i$ . Consequently, using Eqs. (22)–(33), the joint torques  $u_i$  are

$$u_i = \sum_{j=1}^n h_{ij} \ddot{q}_j + u'_i, \quad i=1 \text{ to } n, \quad (34)$$

$$\sum_{j=1}^n h_{ij} \ddot{q}_j = \mathbf{z}_{i-1}^T (\mathbf{n}'_{ia1} + \mathbf{n}'_{ia2} + \mathbf{n}'_{ia3}) \ddot{\mathbf{q}}, \quad (35)$$

$$u'_i = \mathbf{z}_{i-1}^T \mathbf{n}'_{ic}. \quad (36)$$

Herein,  $\dot{q}_i(t_k)$  and  $\ddot{q}_i(t_k)$  are used to calculate  $h_{ij}$  and  $u'_i$ .

The computations needed to solve the linear system of Eq. (34) in order to compute  $\ddot{q}_i(t_k)$  can be performed using Gaussian elimination.

#### IV. NUMERICAL EXAMPLE OF 4-DOFS SPATIAL DISORIENTATION TRAINER

The spatial disorientation trainer (SDT) is designed as a 4-DoFs manipulator with rotational axes, Fig 2. Herein,  $\mathbf{p}_1^* = a_1 [c_1 \ s_1 \ 0]^T$ ,  $\mathbf{p}_2^* = [0 \ 0 \ d_2]^T$ ,  $\mathbf{p}_3 = \mathbf{p}_4 = \mathbf{p}_5 = \mathbf{0}$ .

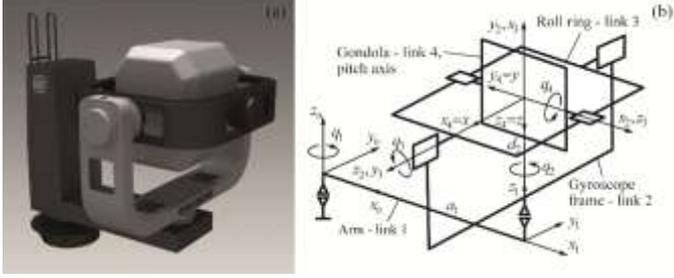


Fig. 2. (a) 3D model of the four DoFs SDT. (b) Coordinate frames of the SDT.

The vectors  $\mathbf{n}'_{ic}$ ,  $\mathbf{n}'_{ia1}$ ,  $\mathbf{n}'_{ia2}$ , and  $\mathbf{n}'_{ia3}$ , Eqs. (22)–(31), for the SDT are

$$\begin{aligned} \mathbf{n}_{4c} &= \mathbf{d}_4 + \mathbf{l}_4 \times \mathbf{e}_4, \quad \mathbf{n}_{3c} = \mathbf{d}_3 + \mathbf{l}_3 \times \mathbf{e}_3, \quad \mathbf{n}_{2c} = \mathbf{d}_2 + \mathbf{l}_2 \times \mathbf{e}_2 + \mathbf{p}_2^* \times \mathbf{E}_3, \\ \mathbf{n}_{1c} &= \mathbf{d}_1 + \mathbf{l}_1 \times \mathbf{e}_1 + \mathbf{p}_1^* \times \mathbf{E}_2, \quad \text{where } \mathbf{E}_3 = \mathbf{e}_3 + \mathbf{e}_4, \quad \mathbf{E}_2 = \mathbf{E}_3 + \mathbf{e}_2, \\ \mathbf{n}_{4a1} &= \sum_{k=1}^4 (\mathbf{d}_{4k} + \mathbf{l}_4 \times \mathbf{e}_{4k}) \ddot{q}_k = \sum_{k=1}^4 \mathbf{n}_{4ka1} \ddot{q}_k, \\ \mathbf{n}_{3a1} &= \sum_{k=1}^3 (\mathbf{d}_{3k} + \mathbf{l}_3 \times \mathbf{e}_{3k}) \ddot{q}_k = \sum_{k=1}^3 \mathbf{n}_{3ka1} \ddot{q}_k, \\ \mathbf{n}_{2a1} &= \sum_{k=1}^2 (\mathbf{d}_{2k} + \mathbf{l}_2 \times \mathbf{e}_{2k}) \ddot{q}_k = \sum_{k=1}^2 \mathbf{n}_{2ka1} \ddot{q}_k, \\ \mathbf{n}_{1a1} &= (\mathbf{d}_{11} + \mathbf{l}_1 \times \mathbf{e}_{11}) \ddot{q}_1 = \mathbf{n}_{11a1} \ddot{q}_1, \end{aligned} \quad (37)$$

$$\begin{aligned} \mathbf{n}'_{4a1} &= [\mathbf{n}_{41a1} \quad \mathbf{n}_{42a1} \quad \mathbf{n}_{43a1} \quad \mathbf{n}_{44a1}] \\ &= [\mathbf{d}_{41} + \mathbf{l}_4 \times \mathbf{e}_{41} \quad \mathbf{d}_{42} + \mathbf{l}_4 \times \mathbf{e}_{42} \quad \mathbf{d}_{43} + \mathbf{l}_4 \times \mathbf{e}_{43} \quad \mathbf{d}_{44} + \mathbf{l}_4 \times \mathbf{e}_{44}], \\ \mathbf{n}'_{3a1} &= \mathbf{n}_{3a1} + \mathbf{n}_{4a1} = [\mathbf{n}'_{31a1} \quad \mathbf{n}'_{32a1} \quad \mathbf{n}'_{33a1} \quad \mathbf{n}'_{34a1}] \\ &= \left[ \sum_{k=3}^4 (\mathbf{d}_{k1} + \mathbf{l}_k \times \mathbf{e}_{k1}) \quad \sum_{k=3}^4 (\mathbf{d}_{k2} + \mathbf{l}_k \times \mathbf{e}_{k2}) \quad \sum_{k=3}^4 (\mathbf{d}_{k3} + \mathbf{l}_k \times \mathbf{e}_{k3}) \quad \mathbf{n}'_{44a1} \right], \\ \mathbf{n}'_{2a1} &= \mathbf{n}_{2a1} + \mathbf{n}'_{3a1} = [\mathbf{n}'_{21a1} \quad \mathbf{n}'_{22a1} \quad \mathbf{n}'_{23a1} \quad \mathbf{n}'_{24a1}] \\ &= \left[ \sum_{k=2}^4 (\mathbf{d}_{k1} + \mathbf{l}_k \times \mathbf{e}_{k1}) \quad \sum_{k=2}^4 (\mathbf{d}_{k2} + \mathbf{l}_k \times \mathbf{e}_{k2}) \quad \mathbf{n}'_{33a1} \quad \mathbf{n}'_{44a1} \right], \end{aligned} \quad (38)$$

$$\begin{aligned} \mathbf{n}'_{1a1} &= \mathbf{n}_{1a1} + \mathbf{n}'_{2a1} = [\mathbf{n}'_{11a1} \quad \mathbf{n}'_{22a1} \quad \mathbf{n}'_{33a1} \quad \mathbf{n}'_{44a1}] \\ &= \left[ \sum_{k=1}^4 (\mathbf{d}_{k1} + \mathbf{l}_k \times \mathbf{e}_{k1}) \quad \mathbf{n}'_{22a1} \quad \mathbf{n}'_{33a1} \quad \mathbf{n}'_{44a1} \right], \end{aligned} \quad (39)$$

$$\mathbf{n}_{4a2} = \mathbf{0}, \quad \mathbf{p}_3^* = \mathbf{0} \Rightarrow \mathbf{n}_{3a2} = \mathbf{0},$$

$$\mathbf{n}'_{4a2} = \mathbf{n}'_{3a2} = [\mathbf{0} \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{0}],$$

$$\mathbf{n}'_{2a2} = [\mathbf{p}_2^* \times \mathbf{E}_{31} \quad \mathbf{p}_2^* \times \mathbf{E}_{32} \quad \mathbf{p}_2^* \times \mathbf{E}_{33} \quad \mathbf{0}],$$

$$\mathbf{n}'_{1a2} = [\mathbf{p}_1^* \times \mathbf{E}_{21} + \mathbf{p}_2^* \times \mathbf{E}_{31} \quad \mathbf{p}_1^* \times \mathbf{E}_{22} + \mathbf{p}_2^* \times \mathbf{E}_{32} \quad \mathbf{p}_2^* \times \mathbf{E}_{33} \quad \mathbf{0}], \quad (40)$$

where

$$\mathbf{E}_{31} = \mathbf{e}_{41} + \mathbf{e}_{31}, \quad \mathbf{E}_{32} = \mathbf{e}_{42} + \mathbf{e}_{32}, \quad \mathbf{E}_{33} = \mathbf{e}_{43} + \mathbf{e}_{33}, \quad \mathbf{E}_{21} = \mathbf{e}_{31} + \mathbf{e}_{21}, \quad \mathbf{E}_{22} = \mathbf{e}_{32} + \mathbf{e}_{22},$$

$$\mathbf{n}'_{2a3} = [\mathbf{0} \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{p}_2^* \times \mathbf{E}_{44}], \quad \mathbf{n}'_{1a3} = [\mathbf{0} \quad \mathbf{0} \quad \mathbf{p}_1^* \times \mathbf{E}_{33} \quad (\mathbf{p}_1^* + \mathbf{p}_2^*) \times \mathbf{E}_{44}], \quad (41)$$

where  $\mathbf{E}_{44} = \mathbf{e}_{44}$ .

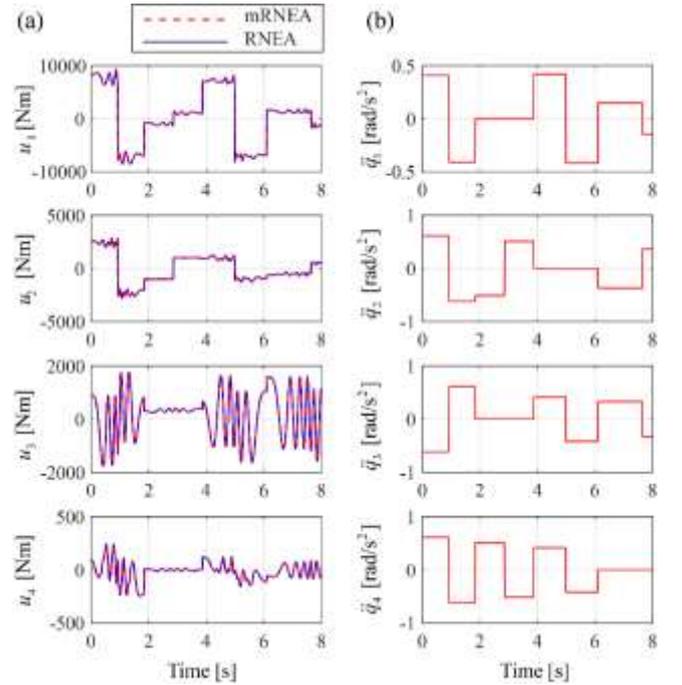


Fig. 3. (a) Link torques  $u_i$  calculated using RNEA and mRNEA. (b) Link accelerations of SDT.

In this numerical example, the consecutive link positions  $q_i(t_k)$ , velocities  $\dot{q}_i(t_k)$ , and accelerations  $\ddot{q}_i(t_k)$  of the SDT are used as the input. Based on these data, and the inertial and geometric parameters of the SDT links, the torques  $u_i$  are calculated using RNEA and mRNEA. The same results were obtained using both algorithms, Fig. 3(a). Consequently, solution of the linear system of Eqs. (34) gave the values of  $\ddot{q}_i(t_k)$  that coincides with the input accelerations, Fig. 3(b).

## V. FD ALGORITHM RELATIVE TO MANIPULATOR ACTUATORS

Similar to Eq. (1), the motion equations for the manipulator relative to the torques/forces of the robot actuators can be written as

$$\mathbf{H}_a(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{u}'_a(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{k}_{ea}) = \mathbf{u}_a, \quad (42)$$

$$\mathbf{u}'_a(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{k}_{ea}) = \mathbf{C}_a(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}_a(\mathbf{q}) + \mathbf{J}(\mathbf{q})^T \mathbf{k}_{ea} + \mathbf{F}_v \dot{\mathbf{q}} + \mathbf{F}_s \text{sign}(\dot{\mathbf{q}}). \quad (43)$$

Herein,  $\mathbf{H}_a(\mathbf{q})$ ,  $\mathbf{u}'_a(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{k}_{ea})$ , and  $\mathbf{u}_a$  relate to the actuator rotors.  $\mathbf{F}_v$  denotes an  $n \times n$  diagonal matrix of viscous friction coefficients  $f_{vi}$ . The static friction torques are considered as Coulomb friction torques;  $\mathbf{F}_s$  is an  $n \times n$  diagonal matrix of the Coulomb friction constants. Herein,  $\text{sign}(\dot{\mathbf{q}})$  denotes an  $n \times 1$  vector whose components are given by the sign functions of single joint velocities.

If certain absolute value of  $u_{ai}$  exceeds its limit, it is reduced to the maximum possible. These achievable values of  $u_{ai}$  are then used in the following linear system of  $n$  equations to calculate the achievable (realistic) joint accelerations  $\ddot{q}_i$ :

$$\sum_{j=1}^n h_{aij} \ddot{q}_j = u_{ai} - u'_{ai}, \quad i = 1 \text{ to } n. \quad (44)$$

## VI. CONCLUSIONS

An algorithm which calculates the achievable joint accelerations in each interpolation cycle based on the FD model and actuators' capabilities was given in this paper. Presented algorithm enables the setting of only attainable joint velocities within each interpolation cycle as determined from the joint acceleration by taking into account the achievable actuator torques. As a result, a precise simulation of the robot movements is provided. This algorithm can indicate to the operator that the programmed parameters of the movements are not achievable. Furthermore, calculation of the realistic forces and moments of the robot joints can be achieved when the simulation system is used in the design phase.

In presented simulation, the mRNEA that gives the mass matrix  $\mathbf{H}$  and the bias vector  $\mathbf{u}'$  of a dynamic model was used. Consequently, mRNEA allows for solving FD by calculating ID only once. It was shown that proposed FD algorithm does not need to calculate the input vector  $\mathbf{u}$  of the FD algorithm, which additionally increases the computational efficiency of the presented method. Compared with the other methods given in the literature, the algorithm presented herein is one of the most efficient ones. Apart from that, it is very simple to develop and implement.

## ACKNOWLEDGMENT

This research is supported by the Ministry of Education, Science and Technological Development of Serbia.

## REFERENCES

- [1] E. Otten, "Inverse and forward dynamics: models of multi-body systems," *Philosophical Transactions of the Royal Society of London, Series B: Biological Sciences*, 358(1437) (2003) 1493-1500.
- [2] B. Siciliano, L. Sciavicco, L. Villani, G. Oriolo, "Dynamics," in: *Robotics: Modelling, Planning and Control*, Springer-Verlag London Limited, 2009, pp. 247-302.
- [3] M. Hirschorn, J. Kövecses, "The role of the mass matrix in the analysis of mechanical systems," *Multibody Syst. Dyn.*, 30 (2013) 397-412.
- [4] R. Featherstone, D. Orin, Dynamics, in: *Handbook of robotics*, B. Siciliano, O. Khatib (Eds.), Springer-Verlag London Limited, 2016, pp. 35-65.
- [5] R. Featherstone, D. Orin, "Robot Dynamics, Equations and Algorithms," *IEEE international conference on robotics and automation (ICRA)*, San Francisco, CA, 24-28 (2000) 826-834.
- [6] M. Walker, D. Orin, "Efficient dynamic computer simulation of robotic mechanisms," *Trans. ASME, J. Dynamic Systems, Measurement & Control*: 104 (1982) 205-211.
- [7] M. Walker, D. Orin, "Dynamics," in: *Robot motion: Planning and Control*, M. Brady, J. Hollerbach, T. Johnson, T. Lozano-Perez, M. Mason (Eds.), MIT Press, 1984, pp. 51-126.
- [8] M. Walker, "Kinematics and Dynamics," in: *Handbook of industrial robotics*, S.Y. Nof (Eds.), John Wiley and Sons, Inc., 1985, pp. 80-95.
- [9] L. Tsai, "Dynamics of Serial Manipulators," in: *Robot analysis: the mechanics of serial and parallel manipulators*, John Wiley and Sons, Inc, 1999, pp. 372-423.
- [10] R. Featherstone, *Rigid Body Dynamics Algorithms*, Springer, 2008.
- [11] J. Hollerbach, "A recursive Lagrangian formulation of manipulator dynamics and a comparative study of dynamics formulation complexity," *IEEE Transactions on Systems, Man, and Cybernetics*, 10 (11) (1980) 730-736.
- [12] V. Kvrđic, J. Vidaković, "Efficient method for robot forward dynamics computation," *Mechanism and Machine Theory* 145 (2020) (10368) 1-25, DOI: 10.1016/j.mechmachtheory.2019.103680.

# Identifying Unmodelled Dynamics in Contact Tasks in Industrial Robotics

Zaviša Gordić, Kosta Jovanović

**Abstract**—The intention of this paper is to propose and discuss different approaches to identifying unmodelled dynamics found in tasks performed by industrial robots. Unmodelled dynamics, forces and torques found in contact tasks cause deviations of joint currents and or torque measurements from their expected values. Correct classification of these deviations and their distinction from collision-induced disturbances is of utmost importance for a reliable collision detection algorithm. Aspects and concerns of different approaches which can help make this distinction are discussed in terms of their reliability, applicability, and versatility. All approaches rely on information which is possible to obtain from industrial robots with closed architecture using standard commands and without any alteration to their control structure.

**Index Terms**— Unmodelled Dynamics Physical Human-Robot Interaction Collision Detection Industrial Robots.

## I. INTRODUCTION

The current situation and future market trends show the increase in demands for more complex, highly customizable and small batch production. Large companies with installed robots and automation are finding it difficult to adapt their production which was suited to low-mix and medium to large batch production, performing repetitive and tedious tasks in areas isolated from human workers because of safety regulations. On the other side, most of customization and small batch production is predominantly performed by human workers in SMEs who are in a constant struggle to keep up with increased demand for higher and more efficient output and reduced costs. The answer to many of the demands is to facilitate interaction of human workers and robots to synergize best of their advantages and achieve unrivalled dexterity, efficiency, and production costs.

The topic of Physical Human-Robot Interaction (pHRI) has been the subject of numerous research, both in field of collision detection [1]-[15] and safety considerations [16]-[18]. One of the main topics within pHRI is the collision detection, as the innermost level of protection which ensures safety of the worker during human – robot collaboration [2],[6]. A lot of research has been done in this field, with very good results in sensitivity and reliability, regardless of whether an industrial

[10]-[12] or a collaborative robot is used [1]-[7]. Most of the related work has utilized model-based collision detection, which requires an accurate model of the robot, but there are also solutions which do not require dynamic model [4]-[5],[12]. While these collision detection algorithms can achieve good performance, most of them were tested in tasks which do not require physical contact of the robot with surroundings. The exception to this pattern can be found in strictly repetitive tasks [12], or in work with detection of intentional human contact through use of filtering [10]-[11],[15]. However, in the latter, the detected intentional contact was detected only on the dynamics of change in robots' joint measurements, and used to bring the robot into compliant state, rather than to continue to identify collisions during a task. In other cases where such contact was allowed, it was predominantly through use of force/torque sensors mounted between the robot tool and the flange of the robot [10]-[11]. The force/torque sensor enables the algorithm to compensate for and to distinguish intentional contact and unmodelled dynamics of the tool and/or load from collisions. However, force/torque sensor mounted between the robot's flange and its end-effector is only able to detect forces/torques at the end of kinematic chain. Therefore, this solution can only be used as a supplement to an existing collision detection algorithm, since collisions may occur at any point between the robot base and its end effector, which would otherwise not be registered by the sensor. Furthermore, such solution is often not suited for the production environment, and therefore robots must rely on their intrinsic sensors.

For industrial robots, which are predominant in production environments, internal sensors include joint position encoders and joint current measurements, based on which joint torques can be estimated. In both cases, detected deviations from the expected values of joint currents/torques originating from intentional contact or unmodelled dynamics can be difficult to distinguish from those originating from collisions, which was recognized in [14]-[15],[19]. This fact combined with necessity to react in shortest amount of time may lead to either high false alarm occurrence frequency or compromised worker safety. Having in mind that the worker safety must be ensured at all times, the effort must be directed into finding a way to reduce the frequency of false collision detections.

The intention of this paper is to propose and discuss different approaches to reaction to unexpected deviations and their proper classification. The main idea is that the intentional deviations can be recorded and integrated into collision detection algorithm, while all deviations different from them should be classified as collisions.

Zaviša Gordić is with the School of Electrical Engineering, University of Belgrade, 73 Bulevar kralja Aleksandra, 11020 Belgrade, Serbia (e-mail: zavisa@etf.rs).

Kosta Jovanović is with the School of Electrical Engineering, University of Belgrade, 73 Bulevar kralja Aleksandra, 11020 Belgrade, Serbia (e-mail: kostaj@etf.rs).

Following a brief introduction to the topic in the First Section, measurement signal properties are discussed in the Second Section. The section presents major attributes related to the nature of the signal and observations based on measurement on an industrial robot performing contact and manipulations tasks.

The Third Section proposes and discusses different approaches to choosing the appropriate signals and ways in which they can be utilised in order to make the distinction between the intentional actions of the robot and the real collisions. The section also discusses how signals can be used to mark the beginning and the end of the deviation as well as how kinematic parameters can be used to refine and improve the decision making.

Conclusion and discussion of the paper is provided in the Fourth Section along with directions for future research.

## II. MEASUREMENT SIGNAL PROPERTIES

Based on the International Robotics Federation, robot installations in 2018 were beyond 400.000 units per year, and most of the robots installed around the world are applied in different fields of industrial production <sup>[20]</sup>. Therefore, vast majority of the installed robots are 5-7 axis articulated or 4 axis SCARA configuration industrial robots. Unlike collaborative robots which are just starting to make entry into manufacturing environments, industrial robots were not initially intended to work with humans in their workspace. To enable them to collaborate with humans, existing sensors must be utilized. The signal based on which collisions are detected is typically a measurement of joint current values. This signal may be available in form of signed or absolute values of joint currents, or as joint torque values estimated using measurements of the current. Consequently, regardless of the form in which the signal is provided, its properties remain the same, and the following analysis is valid for all of them.

One of the most important properties of the signal is related to the time domain repeatability of the current measurement. In previous work related to collision detection <sup>[12],[19]</sup>, it was explained that industrial robots typically do not have a possibility to run parallel processes, but instead emulate this behaviour by swiftly jumping from one routine to another. This fact causes slight deviation of time instants in which identical command will be executed within a repetitive cycle. This also includes the motion commands, which consequently leads to the situation in which measurements from consecutive cycles of the identical movement do not match in time domain, as illustrated on Fig. 1.

The second observation related to the nature of the measurement signal of current/torque is the irregular occurrence of peaks, as shown on Fig. 2. The occasional appearance of peaks in measurement is related to the fact that currents/torques are dependant on speed and acceleration in each joint <sup>[13],[15]</sup>. Due to the fact that industrial robots possess only position measurements from encoders, speed and acceleration are calculated numerically as derivatives and double derivatives of position measurements. Depending on the

actual dynamics of the position change in relation to the sampling rate, these numerical derivations may result in values which do not correspond accurately to the actual dynamics of the robot <sup>[15]</sup>. Although these numerical values may differ from real values, they are absolutely adequate for the proper functioning of the robot due to robot's inherent actuators and structural inertia which filter out the high frequencies of the control signal. However, from the standpoint of observation of signal deviations, these peaks may be mistaken for unexpected external forces. The peaks must not be filtered out because they may contain important indications of a collision or other real external force/torque which are of importance for the collision detection algorithms and worker safety.

Repeatability of the measurement shape and values for identical movements is one of the most important aspects related to the identification of signal deviations. This signal property should be observed from the perspective of the robot itself i.e. its actuators, and from the nature of the desired contact task. At this point, a distinction must be made between deviations caused by unmodelled dynamic properties of the end effector attached throughout the duration of the task on one side and, on the other side, forces/torques exerted or endured by the end effector or dynamics originating from the objects the robot manipulates or interacts with. While the former are generally very easy to determine or automatically estimate even by the built-in functionalities featured in most industrial robots, the latter can be a big issue to quantify and describe in an adequate way.

Previous work has shown that measurement shape and values which originate from the robot itself performing a non-contact task are very repeatable and dependable. However, it was demonstrated that repeatability is not guaranteed for all kinds of contact tasks. For example, while pick and place generally generate predictable and repeatable measurement values for the identical movement, the same cannot be guaranteed for snap fit and spring latch assembly, as shown on Fig. 3 and 4. These two assembly tasks are highly dependent on the repeatability of the object of the assembly and its accurate positioning prior to the assembly process. Different stiffnesses of the springs in spring latch assembly, or slightly altered position of the snap fit object result in significant differences of the measurement signal, as shown on Fig. 3. While the general shape of the deviation from values without contact are similar in shape, the intensities may vary significantly.

From the aforementioned observations, it is possible to understand that industrial robots with closed control architecture pose a unique set of challenges when it comes to detecting deviations from their expected joint current values. When these challenges are viewed from the perspective of collision detection, it should additionally be noted that all signal processing and decision making must be made in very short amount of time in order to make timely reactions of the robot and prevent injury or damage.

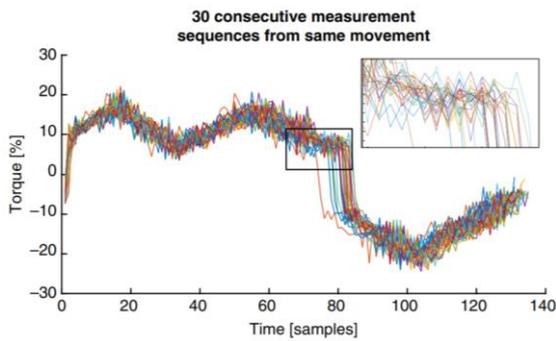


Fig. 1. Measurements from 30 consecutive executions of the same movement<sup>[12]</sup> show time domain related differences in recorded signal caused by the differences in sampling instants. Signals have slightly different lengths, and they are shifted in time for up to 13 samples or just under 10% of the total length of the signal.

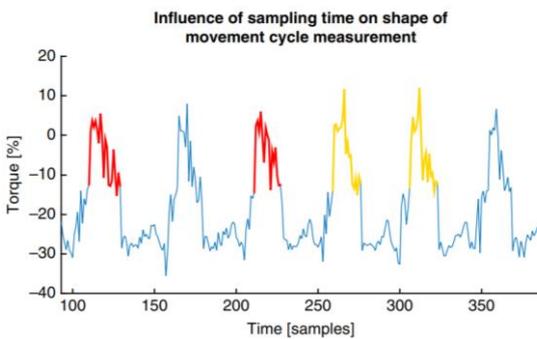


Fig. 2. Unpredictable occurrence of peaks due to differences in sampling instants<sup>[12]</sup>. The six shown periods of the signal represent six successive cycles of the same motion task. Peaks marked in yellow and red show biggest effects of the issues related to numerical position derivation.

### III. SUGGESTED APPROACHES

The observations from previous section lead to a conclusion that some types of contact tasks are difficult to predict in their intensity and shape, and therefore difficult to distinguish from collision-induced deviations and numerical anomalies. Additional difficulty is that distinction must be made with first samples of deviation in order to minimize the consequences of potential collision.

The presumption for all approaches presented in this paper is that robot performs more or less repetitive tasks in cycles, which is the case in vast majority of tasks for industrial or collaborative robots alike. Therefore, the idea is to record the deviation(s) occurring in one cycle of the robot task while operating in a controlled and supervised environment without collision occurrence. The recorded deviation can be used to extract and generalize certain features of the cycle that can be used as indicators of when and in which shape and intensity to expect deviation originating from an intentional and desired action of the robot in future cycles. With this approach, any deviation from the values of the recorded deviation should be considered as a collision.

To be able to distinguish unmodelled dynamics and forces/torques caused by intentional contacts from those originating from a collision, it is important to know two

things. First – the shape and intensity of deviation originating from the unmodelled action, and second – when to expect it and for how long.

To aid in this matter, it is needed to consider which information from the robot, and in which capacity, can be used to complement the measurements of currents/torques.

When various brands and generations of industrial robots, as well as the general logic behind control algorithms are considered, other signals readily available on industrial robots can be identified.

One widely available signal that can be used to aid in this matter is the position measurement either in joint space or in external space, mostly described in Cartesian frame. For robots which can provide only joint position measurements, it is possible to use an automatic procedure to identify kinematic parameters<sup>[21]</sup> of the robot and provide external frame coordinates using direct kinematics calculations. Position and/or orientation of the flange in external frame, the operating plane, or any other of kinematics-related parameters can have a potential to be used as an indicator of when to expect the start and/or end of deviation. Even more importantly, kinematic parameters can be placed into context of estimating the changes in overall shape of the deviation.

Another option is to use output signals used by the robots to interact and control external entities. These signals can be of good use if it is directly and unambiguously related to the action which causes deviation. One such example can be the output which controls the gripper, since as long as the gripper is closed, the load is connected to the robot, and its influence is present. From the moment the gripper is opened, the load is released, and the deviation is not expected.

The aforementioned types of signals can be used effectively not only to help predict the start and end of the deviation, but also to assist in determining the changes of the shape of deviation caused by the change in robot's posture.

When it comes to the shape of deviations, the recorded deviation is only valid for the identical movement to the one during which it was recorded. Changing the robot movement, the weight distribution of the manipulated object or some other parameter from the cycle in which the deviation was recorded would result in changes to its shape, as illustrated on Fig. 3 and 4. While the overall shape is similar, some levels, peaks or other shape features may be different. To enable matching of the recorded deviation with the one under new conditions, it is needed to predict how it would change.

One way of determining the possible variations to the shape of deviation caused by the possible differences between the cycles of the task execution is to use mostly analytical approach. Forces and/or torques affecting the robot while performing contact tasks can generally be divided into tasks where predominant deviations from the expected values originate from weight and inertia of the manipulated object and those whose origin can be related to the forces/torques the robot itself and/or robot's end-effector exert on an external entity.

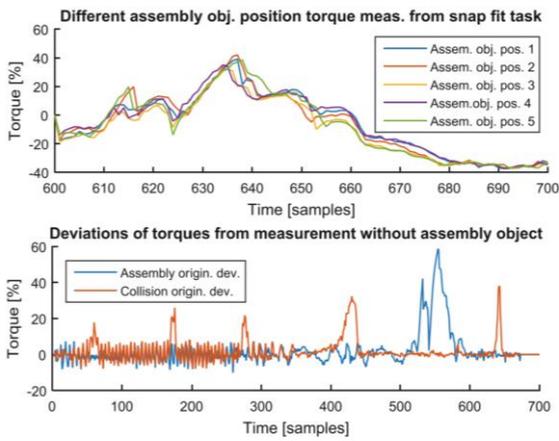


Fig. 3. Deviations of the torque measurement during snap fit assembly task<sup>[19]</sup>. (upper) Differences in positioning of under 1mm caused that some deviations have additional peaks at sample time 615. (lower) Deviation signal from the snap fit assembly task compared to collision-induced deviations shows great similarity.

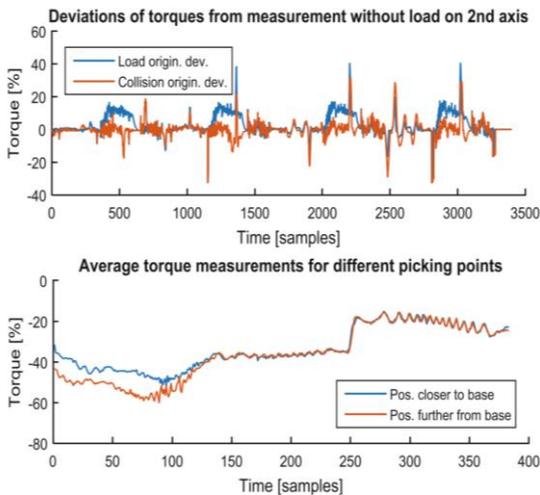


Fig. 4. Deviations caused by unmodelled load during manipulation task<sup>[19]</sup>. (upper) The deviations of the torque measurement due to unmodelled load compared to collision-induced deviation. While the nature of deviations is different, the intensity can in some cases be comparable. The first two periods of the signal correspond to cycles with picking points closer to the robot's base, while second two periods correspond cycles in which picking points were further away. (lower) Differences in distance of the picking points from the robot's base has caused differences in part of the deviation. From time sample 142, the deviations start to match because that part of the trajectory was common to both cycles of the repetitive task.

For the first group of forces/torques, the spatial configuration of the robot and dynamics of its transition to another configuration is the predominant factor which influences the shape and intensity of the deviation. When the robot is stationary, these deviations are constant, with exception to short transient periods, for example in liquids container manipulation, and reflect onto non-vertical axis. Tasks which belong to this group include typically manipulation actions such as pick and place, loading/unloading, machine tending, palletizing, and packaging.

For the second group, the direction of the exerted force/torque is relative to the orientation of the flange of the robot. The shape and intensity of the deviation is projected

onto individual robot joints depending on their orientation relative to the flange, while its dependence on the robot movement is negligible. Robot tasks which belong to this group include screwing, drilling, polishing, riveting, various types of assembly without adhesives or material depositing, probing, friction welding etc.

There are tasks which are a combination of the previous two groups. Most of these comprehensive tasks can be divided into smaller sub-tasks, in order to subject them to the aforementioned division. For tasks where decoupling according to the division is not feasible, the deviation will be a resultant of the individual deviations from the two groups.

With the aforementioned considerations and division in mind, it is possible to understand how the recorded shape of the deviation may evolve due to changes in robot movement and posture. For the analytical approach, it is necessary to use knowledge of robot kinematics to reconstruct the general direction and intensity of the external force/torque relative to any relevant section of the robot. Then, it would be needed to calculate how it would project on individual joints in another configuration, and how the dynamics of the robot manipulator would affect it. If estimation of the changes of the deviation shape and its projection is not feasible, it is possible to use measures of curve similarity, as presented in<sup>[22]-[25]</sup>, or with some modifications, mostly related to real-time application. The positive side of the analytical approach is that it is a verifiable and understandable and transparent process, which is a desirable trait for all matters related to collision detection. Furthermore, the process does not need a vast training set, or a lot of time to set up. The downsides are its complexity, and the fact that it is difficult to completely automate or make user friendly for operators with non-expert knowledge.

An important aspect related to the analytical approach is that it requires additional logic to determine when the deviation is expected to occur, and when to end. This information can in some cases be extracted by observing the kinematic parameters. As mentioned earlier, it is necessary to find a good correlation and a certain repetitive pattern in kinematic parameters such as constant height, position or orientation, geometrical surface/line to which the robot arrives just before the action which causes the deviation starts or just before it ends, This may prove to be an extremely challenging matter, considering the entire spectrum of tasks that a robot may be required to perform, and having in mind that setup phase should be as brief as possible. Of course, if the number of different positions to which a robot should arrive to perform the action that causes the deviation is finite and known in advance, this information could be incorporated into the algorithm.

As mentioned earlier, observing the output signals of the robot or other devices that trigger the actions causing deviations is available, it should be taken into consideration whenever possible.

Another approach to estimating the changes in shape of the recorded deviation would be to use Artificial Neural Networks (ANN). ANNs are able to process a large amount of input data and to find connections and correlations that work well, and

that would otherwise remain unused. Research <sup>[4],[25]</sup> has shown that ANNs can be quite successfully trained to extrapolate and predict joint currents/torques of the robot performing new tasks. This can be done with or without knowledge of kinematic parameters of the robot, and with various other information provided by the robot itself or some of the external entities related to the action that causes the deviation. Following the same logic, it is reasonable to assume they could predict not only the shape of deviation, but maybe also when to expect it, and for how long. To the best knowledge of the authors, the latter has not been a subject of research in any related field, and therefore it can only be assumed based on the general capabilities of ANNs. The problem to this approach, however, is common to many tasks involving ANNs, and that is training. Generally speaking, ANNs require a lot of input data, and a lot of time to process all information and make appropriate connections. For all the material needed for the training, supervised measurements must be made, which is at least impractical from the implementation and exploitation point of view. Moreover, having in mind the high variability of possible contact tasks and types of external contact forces, it may be difficult to be certain of whether a representative training set has been acquired. Although it was shown in <sup>[4],[25]</sup> that the desired signals originating from the robot itself and all permanently attached entities can be predicted with high certainty, generally speaking the task space is quite unique for each individual application, and the variations in external forces must be well studied for each application. Finally, the biggest issue with ANNs is that the user is never absolutely sure how certain values are predicted, and how reliable they are. For collision detection, this is a huge problem, since reliability must be ensured and provable.

It is evident that both the analytical and ANN approach have prospects and concerns, some of which are complementary. A hybrid approach might be the needed solution which would combine the transparency and verifiability of the analytical approach with the ANNs' flexibility and ability to find hidden connections between inputs. Kinematics model of the robot combined with the current/torque measurements in joints can be used to reconstruct the profile of the deviation in Cartesian space. After the shape of deviation is determined, it can be described as a parametrized curve. Points of interests on this curve can be determined from several samples using Eigenvalues of the measurement set. Then, algorithm such as Active Shape Modelling can be used in conjunction with ANNs to determine the relation of the kinematic parameters with the shape of the deviation curve. Simultaneously, ANNs can find the best indicators of when the deviation may occur within the cycle. The algorithm could evaluate its estimation and indicate to the operator when the satisfactory level has been reached. Even after the adequate success rate of the estimations has been reached, the estimation could get better over time with increase of training and evaluation samples.

Hypothetically, the hybrid model could offer a good solution to the problem of distinction between intentional

contacts and collisions. Potentially, it could be improved by incorporating a model of a collision-induced deviation. However, collisions come in different shapes and intensities, and so do the contact task-induced deviations, and there is not a single solution for all tasks. Nevertheless, the proposed approach offers another way to overcome the issues of false alarms occurring in robot contact tasks with unmodelled dynamics and forces/torques.

#### IV. DISCUSSION AND CONCLUSIONS

Joint work of human workers and robots is a logical and desirable trend with increasingly more foothold in reality. Enabling industrial robots with closed control architecture to adapt to current and future needs for pHRI would enable faster adoption of robotization primarily in manufacturing SMEs and provide them the tools to be more competitive and productive. The main precondition to enabling robots for pHRI is to enable them to detect collisions with their surroundings and therefore prevent human injury. One of major obstacles in efficient and reliable collision detection is telling apart intentional, but unmodelled dynamics and forces/torques occurring in contact tasks from real collisions. As a contribution to this goal, this paper suggested and discussed approaches to making this distinction, regardless if it is applied combined with model or non-model-based collision detection algorithms. Although intended for industrial robots as implementation platforms, suggested approaches can be implemented also on collaborative robots, regardless of their configuration and number of axis.

The analysis of the signal properties of currents/estimated torques, based on which the decision making is predominantly made, pointed out some of the main attributes and issues from the measurement point of view. The signals were observed and discussed from the perspective of their dynamics, repeatability, and similarity with collision-induced deviations. Measurement results from <sup>[19]</sup> were used to demonstrate signal properties from various contact and manipulation tasks. They demonstrated how unmodelled load, distance from the robot base and positioning repeatability of parts manipulated by the robot influence on the measured signal.

To achieve better distinction between intentional contact tasks and collisions, additional signals were considered to complement the current/estimated torque signal.

As a most widely available and useful signal, joint or external frame positions can be used to indicate when the section of the task which contains unmodelled dynamics and/or forces/torques will start and end. This functionality requires some previous knowledge which can be obtained through generalization of performing the desired task with human supervision. Both robot flange/end effector position and orientation as well as joint positions can be used to make the generalization. The generalizations can be made using different parameters and making the correlation between them. Parameters which show greatest level of correlation between the different cycles of task execution can be used as good indicators of when to expect the deviations.

Furthermore, it was suggested that with prior knowledge of the kinematic parameters of the robot, it is possible not only to indicate when the unmodelled part of the task will start and end, but also to approximate how the deviation from expected values will evolve depending on the robot joint movement. To this end, three different approaches to estimating the shape of the deviation were considered in general.

The first approach was based on analytics, and its general properties were discussed without going into particularities, along with its advantages and disadvantages. It was pointed out that knowledge of robot's kinematic parameters can also be used to determine the correlation of the robot flange orientation and orientation of the axis most affected by the deviation. The approach based on ANN was also generally discussed in terms of its applicability in collision detection and its prospects and concerns. Finally, it was suggested that a hybrid method might be the best solution for the application in the relevant field.

It was noted that outputs used to activate/deactivate external tools or devices were suggested to be used in tasks where they are available and directly related to the desired tasks. These signals can be used to indicate start of a task section with unmodelled forces/torques and possibly influence the thresholds for collision detection. In many cases, such signals can be used to mark the ending of the task section with unmodelled dynamics and restore detection thresholds.

The considerations and discussion presented in this paper were presented as a general guideline for addressing the aspect of pHRI which is often neglected, yet very important for good quality collision detection. Future work will involve further elaboration and design of deviation detection, estimation and distinction rules in accordance to results presented in this paper and previous research<sup>[12],[19],[21]</sup>.

#### ACKNOWLEDGMENT

The research was partially supported by the Serbian Ministry of education, science, and technological development, and by the Science Fund of the Republic of Serbia, PROMIS, #6062528, ForNextCobot.

#### REFERENCES

- [1] A. De Luca, A. Albu-Schäffer, S. Haddadin and G. Hirzinger, "Collision Detection And Safe Reaction With The DLR-III Lightweight Manipulator Arm," IEEE/RSJ International Conference on Intelligent Robots and Systems – IROS, Beijing, China (2006) pp. 1623–1630.
- [2] A. De Luca, F. Flacco, "Integrated Control for pHRI: Collision Avoidance, Detection, Reaction and Collaboration," IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechanics – BioRob, Roma, Italy (2012) pp. 288–295.
- [3] F. Dimeas, L. Avendaño-Valencia, N. Aspragathos, "Human – robot collision detection and identification based on fuzzy and time series modelling," *Robotica* 33(9), 1886–1898 (2015).
- [4] A.-N. Sharkawy, P. Koustoumpardis, N. Aspragathos, "Manipulator Collision Detection and Collided Link Identification based on Neural Networks," International Conference on Robotics in Alpe-Adria Danube Region – RAAD, Patras, Greece (2018).
- [5] A. De Luca, R. Mattone, "Sensorless Robot Collision Detection and Hybrid Force/Motion Control," IEEE International Conference on Robotics and Automation – ICRA, Edmonton, Alberta, Canada (2005) pp. 1011–1016.
- [6] F. Flacco, A. De Luca, "Safe Physical Human-Robot Collaboration," IEEE/RSJ International Conference on Intelligent Robots and Systems – IROS, Tokyo, Japan (2013) p. 2072.
- [7] E. Magrini, F. Flacco, A. De Luca, "Control of Generalized Contact Motion and Force in Physical Human-Robot Interaction," IEEE International Conference on Robotics and Automation – ICRA, Seattle, Washington, USA (2015) pp. 2298–2304
- [8] M. Erden, T. Tomiyama, "Human-intent detection and physically interactive control of a robot without force sensors," *IEEE Trans. Robot.* 26(2), 370–382 (2010).
- [9] P. Pastor, L. Righetti, M. Kalakrishnan, S. Schaal, "Online Movement Adaptation Based on Previous Sensor Experiences," IEEE/RSJ International Conference on Intelligent Robots and Systems – IROS, San Francisco, CA, USA (2011) pp. 365–371.
- [10] M. Geravand, F. Flacco, A. De Luca, "Human-Robot Physical Interaction and Collaboration Using an Industrial Robot With a Closed Control Architecture," IEEE International Conference on Robotics and Automation – ICRA, Karlsruhe, Germany (2013) pp. 4000–4007.
- [11] E. Mariotti, E. Magrini and A. D. Luca, "Admittance Control for Human-Robot Interaction Using an Industrial Robot Equipped with a F/T Sensor," *International Conference on Robotics and Automation ICRA*, Montreal, QC, Canada, 2019, pp. 6130–6136
- [12] Gordić, Z., & Jovanović, K. (2019). Collision Detection on Industrial Robots in Repetitive Tasks Using Modified Dynamic Time Warping. *Robotica*, 1-20.
- [13] M. Makarov, A. Caldas, M. Grossard, P. Rodríguez-Ayerbe and D. Dumur, "Adaptive Filtering for Robust Proprioceptive Robot Impact Detection Under Model Uncertainties," in *IEEE ASME Transactions on Mechatronics*, vol. 19, no. 6, pp. 1917–1928, Dec. 2014
- [14] C. Pengfei, Y. Gan and X. Dai. "Model-based sensorless robot collision detection under model uncertainties with a fast dynamics identification." *International Journal of Advanced Robotic Systems* 16 (2019)
- [15] Chang-Nho Cho, Joon-Hong Kim, Young-Loul Kim, Jae-Bok Song & Jin-Ho Kyung (2012) Collision Detection Algorithm to Distinguish Between Intended Contact and Unexpected Collision, *Advanced Robotics*, 26:16, 1825-1840
- [16] Y. Yamada, Y. Hirasawa, S. Huand, Y. Umetani, "Fail-Safe Human/Robot Contact in the Safety Space," IEEE International Workshop on Robot and Human Communication – RO-MAN, Tsukuba, Ibaraki, Japan (1996) pp. 59–64.
- [17] S. Haddadin, A. Albu-Schäffer and G. Hirzinger, "Safe Physical Human-Robot Interaction: Measurements, Analysis and New Insights," In *Robotics Research, Springer Tracts in Advanced Robotics* (M. Kaneko and Y. Nakamura, eds.), vol. 66 (Springer, Berlin, Heidelberg, 2010) pp. 395–407.
- [18] S. Haddadin, A. Albu-Schäffer and G. Hirzinger, "Safety evaluation of physical human-robot interaction via crash-testing," In: *Robotics: Science and Systems* (MIT Press, Cambridge, Massachusetts, 2007) pp. 217–224.
- [19] Gordić, Z., Jovanović K. (2020) Influence of Unmodelled External Forces on the Quality of Collision Detection. In: Berns K., Görges D. (eds) *Advances in Service and Industrial Robotics*. RAAD 2019. *Advances in Intelligent Systems and Computing*, vol 980. Springer Cham
- [20] International Federation of Robotics, Executive Summary World Robotics 2019 Industrial Robots, World Robotics 2019 edition, 2019.
- [21] Gordić, Z., Jovanović K. Partial Pose Measurements For Identification Of Denavit-Hartenberg Parameters Of An Industrial Robot, Proceedings of the 4th International Conference on Electrical, Electronic and Computing Engineering (IcETRAN 2017), pp. ROI1.6.1-ROI1.6.4, Kladovo, Serbia, Jun, 2017.
- [22] T.F. Cootes, C.J. Taylor, D.H. Cooper, J. Graham, Active Shape Models-Their Training and Application, *Computer Vision and Image Understanding*, Volume 61, Issue 1, Pages 38-59,1995,
- [23] M. Vlachos, G. Kollios and D. Gunopulos, "Discovering similar multidimensional trajectories," *Proceedings th International Conference on Data Engineering*, San Jose, USA, 2002, pp. 673-684
- [24] T. Nakamura, K. Taki, H. Nomiya, K. Seki, K. Uehara A shape-based similarity measure for time series data with ensemble learning. *Pattern Analysis Applications*, 16, 535–548 (2013).
- [25] J. de Vries, Object Recognition: A Shape-Based Approach using Artificial Neural Networks, University of Utrecht, Department of Computer Science, January 2006

# Impedance Control of an Industrial Manipulator

Jovan Šumarac, Aleksandar Rodić and Kosta Jovanović

Force control has increasingly found its uses in industry as well as been a focus of academic research. Since its proposal in impedance control has become one of the most studied and applied fields of force control. The goal of this paper was to implement an impedance controller and use it for task space control of an industrial robot with six degrees of freedom. The controller model was realized and simulation was done in Matlab Simulink with the industrial R robot as a case study. The robot was given tool position and forces as control inputs. Position and force responses are presented using computer simulation positions were calculated and forces were estimated based on the impedance control law.

**Index Terms**—Impedance control, contact force, Industrial manipulator

## I. INTRODUCTION

Force control is increasingly applied in modern robotics, and it is gaining further momentum as a result of advances in electronics, computer power, and especially force and torque sensors. It is one of the key technologies for integrating robots into human or unknown environments. Its main advantage is making the robot react as soon as it detects an obstacle, preventing damages to itself and/or its surroundings. It is also used for tasks in which there is interaction between the robot and its environment and therefore it is more important to control the force applied by the end effector rather than just its position. Some examples of such tasks are finishing, welding, drilling and the more sophisticated precision assembly tasks, surgery assistance etc.

On the other hand, the classic position control is still the predominant way of programming industrial robots. Typically, the end effector tool follows a prescribed trajectory in space which has been pre-programmed or “taught” before run-time. This type of control is suitable for routine tasks, in which robot surroundings are completely known and no workspace changes or obstacles are expected.

So other approaches were developed over the years, many focusing on combined control of both position and force. One such technique is called impedance control. Since its proposal by Neville Hogan [1][2], impedance control has become one of the most prominent force control fields and arguably one of the most successful.

Jovan Šumarac is with the Robotics Laboratory, Institute “Mihailo Pupin”, University of Belgrade, Volgina 15, 11060 Belgrade, Serbia (e-mail: jovan.sumarac@pupin.rs)

Aleksandar Rodić is with the Robotics Laboratory, Institute “Mihailo Pupin”, University of Belgrade, Volgina 15, 11060 Belgrade, Serbia (e-mail: aleksandar.rodic@pupin.rs)

Kosta Jovanović is with the School of Electrical Engineering, University of Belgrade, 73 Bulevar kralja Aleksandra, 11020 Belgrade, Serbia (e-mail: kostaj@etf.bg.ac.rs)

Although it has been researched and used in innovative applications impedance control can be a very complex task to achieve and new studies have continuously been published since its proposal. Authors have applied it to robot human interaction systems [3], systems with collaboration between two robots [4] as well as for simulation of human muscle movements [5]. It shows that this technique and its integration into various robotic systems are still of academic interest.

This paper presents a concise explanation of the impedance control algorithm and its application on an industrial robot. The robot in question is a typical serial manipulator with six revolute joints and six degrees of freedom. The controller and the system were modeled, simulated and the results presented and analyzed.

## II. IMPEDANCE CONTROL

The hypothesis of paper [1] which originally proposed impedance control is that no controller can prevent the robot from appearing as a physical system to its environment. As a consequence a dynamic interaction between the two must exist. By analogy with electric circuit theory which defines impedance as the voltage/current ratio, mechanical impedance is defined as the ratio between the end effector force and its position. By controlling that ratio the idea is to implicitly control the resistive force of the environment i.e. the dynamic interaction between the robot and the environment. The general strategy is often to control the robot’s position as well as to give it a disturbance response in the form of impedance.

The difference between this approach and the conventional position control is that the impedance controller seeks to control the dynamic ratio between force and position instead of independently controlling one of those variables.

As task space (Cartesian) control is a well-developed technique, modelling the environment with dynamic parameters is simplified when done in Cartesian coordinates. The main problem is which control law to adopt. Since linear control has many advantages and simplifies the problem significantly feedback linearization is an appropriate choice. It is applied in this case as a linear control law of the second order modeled on a mass-spring-damper system. It is also a multivariable control law since the controller is applied to a 6 DoF (degree of freedom) robot and so there are  $n$  control variables. To comply with the given task the desired impedance of the robot end effector is given as:

$$d(\ddot{\mathbf{x}} - \ddot{\mathbf{x}}_d) + a(\dot{\mathbf{x}} - \dot{\mathbf{x}}_d) + a(\mathbf{x} - \mathbf{x}_d) = \mathbf{e}. \quad (1)$$

Where  $d$ ,  $a$  and  $a$  are constant, diagonal, positive-definite  $n \times n$  matrices representing the desired inertia, damping and stiffness system matrices. Vectors  $\mathbf{x}_d$  and  $\mathbf{x}$

represent the desired and actual end effector positions. Vector  $e$  represents the generalized interaction force between the robot and its environment. The variables are in bold font to indicate that they are matrices and vectors rather than scalar values.

The basic idea of impedance control is that if the robot can follow the acceleration reference given by:

$$\ddot{\mathbf{x}}_r = \ddot{\mathbf{x}}_d + \mathbf{d}^{-1}(-\mathbf{e} + \mathbf{d}(\dot{\mathbf{x}}_d - \dot{\mathbf{x}}) + \mathbf{d}(\mathbf{x}_d - \mathbf{x})). \quad (2)$$

then it will behave as described by (1). [1][6] So  $\ddot{\mathbf{x}}_r$  is the reference signal for the acceleration control loop that linearizes and decouples the manipulator's non-linear dynamics. Such a controller will seek to follow the position reference. While the tracking is good, with small position error values the resulting force will tend to return to zero. If the error increases, the interaction force will increase as well, in order to make the controller move the robot in a direction that will decrease the error.

The force vector in (1) can have different interpretations. In case of (1) the controller will seek to minimize any interaction force and that force will be close to zero if the position tracking is good. However if there is a desired contact force value for a certain robot task, then that value can be given to the controller [7]. In that case the force vector would have a value of  $\mathbf{d}^{-1}\mathbf{e}$ , the index  $d$  signifying a desired value. A practical realization of such a controller would mean having a force sensor that could measure and obtain the actual end effector forces. They could be indexed with  $s$  to imply sensor values. In this case the controller would seek to maintain the desired force value.

A key problem of impedance control is calculating the  $\mathbf{d}$ ,  $\mathbf{d}$  and  $\mathbf{d}$  matrix values. As stated, they represent the modeled environment inertia, damping and stiffness. There is often a great deal of uncertainty when modelling and estimating these parameters. They are usually estimated by estimating one by one matrix, while the rest remain constant [8]. Since these are diagonal matrices similar values are usually used for each direction, or at least one set of similar values for linear movement directions and another for angular. The inertia matrix  $\mathbf{d}$  affects the system's response speed. If lower values are chosen the resulting response will be faster, but it will also result in large tool acceleration values. The opposite is true for larger matrix values.  $\mathbf{d}$  matrix models the system damping and therefore is mostly significant during the transient state. If it were equal to zero matrix the transient response time would be infinite, and the system would be completely undamped. As the matrix values increase the system becomes more damped and the transient time is shorter. Finally  $\mathbf{d}$  models the environment's stiffness and has the greatest influence on the tracking itself. With greater matrix values the tracking is better, and the achieved positions and forces are closer to their references.

However, too large values for these matrices can result in too big acceleration values and compromise the system's stability. That is another very important factor in choosing the matrix values. Looking at (1) it is possible to analyze its transfer function in Laplace domain. The admittance, e.g. the position error-tool force ratio has a transfer function

given by:

$$(\mathbf{s}) = \frac{1}{\mathbf{d}\mathbf{s}^2 + \mathbf{d}\mathbf{s} + \mathbf{d}}. \quad (3)$$

If the parameters of these matrices on the main diagonal are all positive, then those matrices are positive-definite. It has been shown that in that case the controller will be stable too [6]. If the model of the controlled robot is stable as well and the controller matrix values are not too large as to affect it, then the stability of the whole system will not be compromised.

This is the guide to choosing the matrix values. Although some proposed tuning methods and the order of magnitude for these values can be found in literature [4] this tuning process is often based on the researcher's experience, as it depends heavily on the type of robot and especially its environment.

The structure of a conventional impedance controller is shown on figure 1:

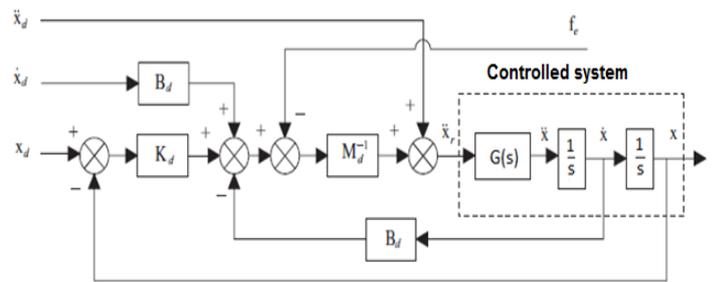


Fig. 1. The structure of a typical impedance controller.

So the controller is realized as given by (1) and the controlled system is added. Sometimes, if there is not an accurate enough dynamic model, the system can be represented just as a double integrator, in which case the  $G$  matrix is equal to an identity matrix of the order of the system. However since the dynamic model of the UR5 robot used in this paper is known,  $G$  function used in this case represents the robot's dynamic model, and will be more detailed in the next chapter.

### III. SIMULATION AND RESULTS

The robot and the controller were modeled and the simulation was done in Matlab/Simulink. In recent years Matlab's Robotics Toolbox package has added models of many commercially available robots including UR5. A large number of ready-made functions are available including the computing of forward and differential kinematics, calculating dynamic equation matrices, trajectory planning, robot 3D animation, etc.

The control scheme given in figure 1 has been slightly altered for Matlab implementation. Since a very good robot model is implemented in Matlab, that model is used instead of the  $G$  function in figure 1. Its inverse and forward dynamics were calculated, and implemented in Simulink with S-function blocks. Also, since the controller is made for task space control and the dynamics calculated in joint

space, kinematics blocks were used for transformations from one space to another. The final scheme is given by figure 2:

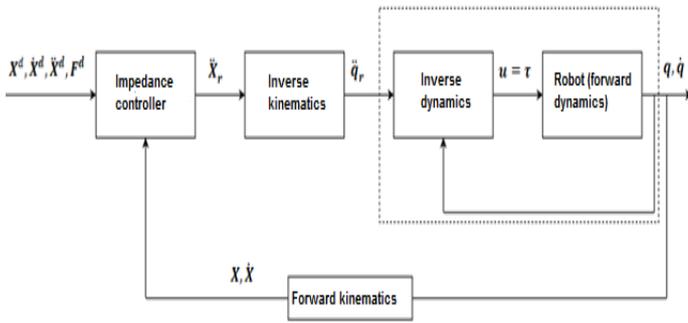


Fig. 2. Impedance control scheme used in simulation.

So, the controller gets the task space coordinates which include the desired tool position, velocity and acceleration as well as the desired contact forces. After getting the controlled values, they are converted to joint space in an inverse kinematics block and given to the robot model. Joint torques are calculated, and joint coordinates are then obtained via forward dynamics. They are returned to the controller after being converted to task space coordinates in a forward kinematics block.

Simulation results are shown in figures 3-8. It should be noted that the position values were obtained in the simulation itself, and shown as such. Force values were then estimated using those values, from the impedance control law as described by (1).

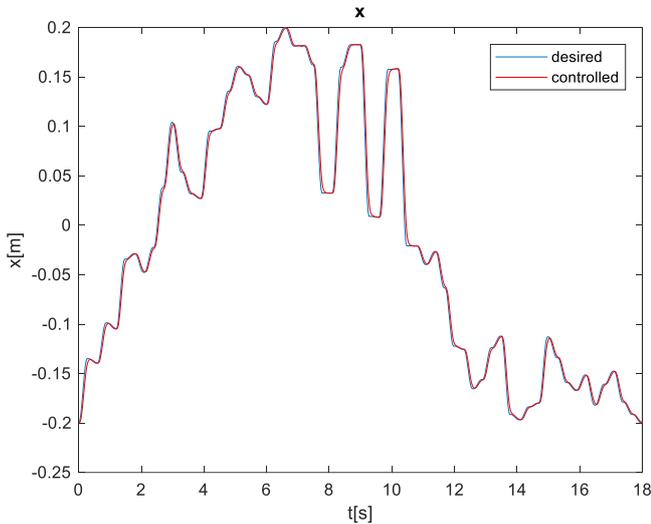


Fig. 3. x coordinate of the tool.

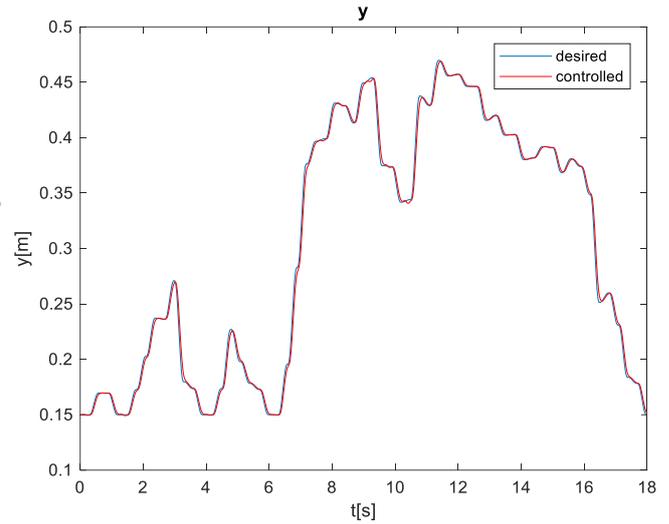


Fig. 4. y coordinate of the tool.

Figures 3 and 4 show  $x$  and  $y$  tool coordinates, respectively. The reference signal in those cases was not a constant value. Rather the signals are constantly changing in a range of approx. 30 to 40 centimeters. The goal was to see how well the controller can follow such signals. It can be seen that there is a very good tracking, the controller can react to the changes relatively quickly and the error is within 2 per cent margin.

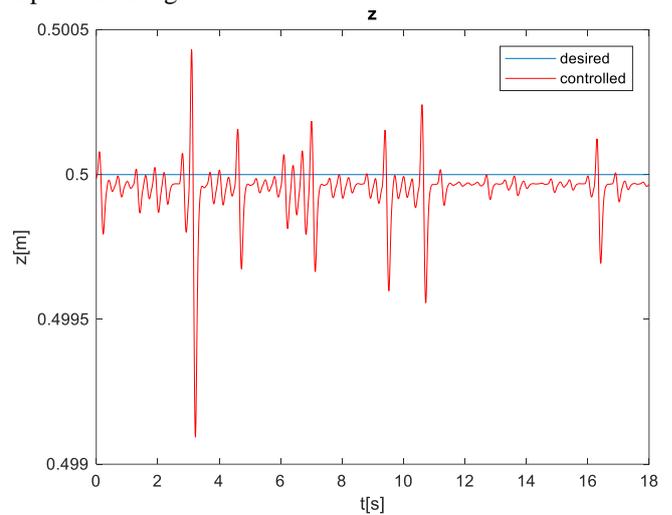


Fig. 5. z coordinate of the tool.

Figure 5 shows the tracking of a constant reference. The tracking is very good in this case too, with the error well below the 2 per cent margin.

The same is true for angular components. They represent the tool's orientation and they were given zero references.

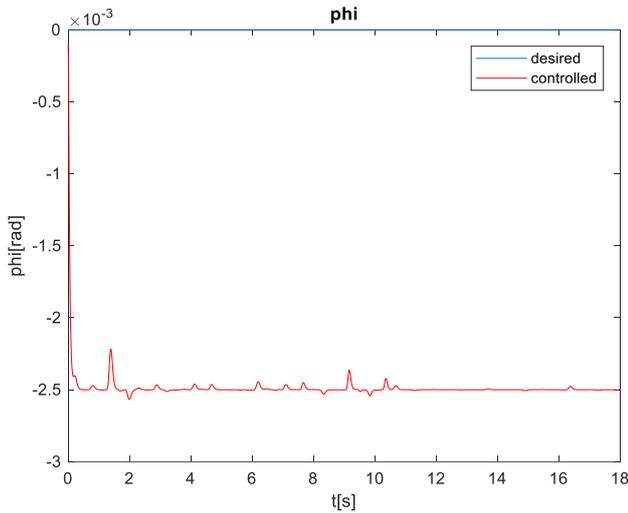


Fig. 6.  $\varphi$  coordinate of the tool.

Figure 6 shows good tracking but also a constant error. The error margin is very small, 0.25 per cent. It is a compromise on the part of the controller in order to satisfy the desired impedance. This is also the case for the remaining two coordinates,  $\psi$  and  $\theta$ . Since the results are very similar, they are shown in a joint, smaller figure.

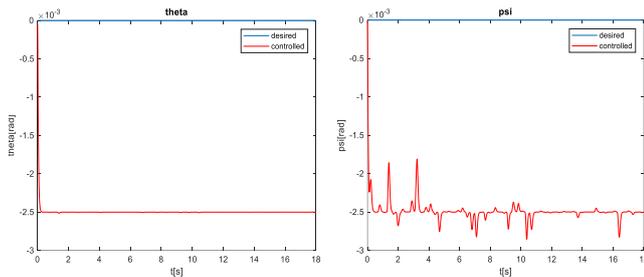


Fig. 7.  $\psi$  and  $\theta$  coordinates of the tool, respectively.

Figures 3 to 7 have shown very good position control. These values were obtained directly from the simulated model and have confirmed the expectations of the impedance control law given by (1). As stated, the forces are now also expected to behave as described by this law.

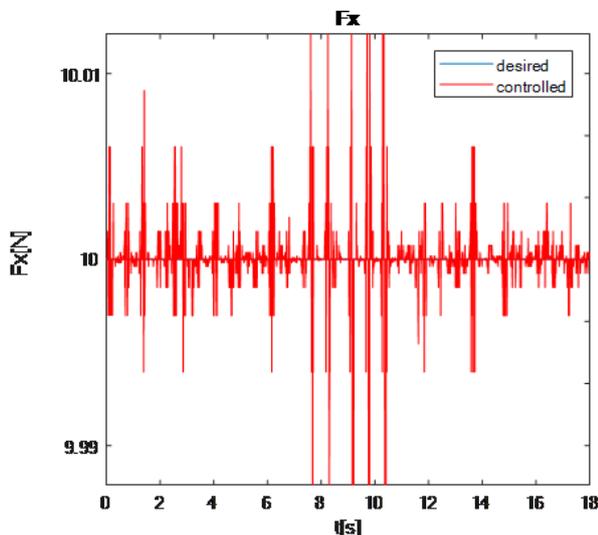


Fig. 8.  $F_x$  coordinate of the tool.

Precise tool force estimation is a complex task that in practice requires knowledge about the robot, its technical specifications (e.g. motor currents or other variables), its environment etc. In this case force values were formed as in (1), after obtaining the actual values for tool position, velocity and acceleration. Since the values are very similar for all six directions of the force, only the force component along the  $x$ -axis is shown here.

Figure 8 shows that the expected result for  $F_x$  is equal to the desired value (10N), which again confirms (1) and (2).

#### IV. CONCLUSION

The results obtained in this simulation were close to the expected ones and have shown a good impedance controller. Position values, which were obtained directly from the simulation, have confirmed a very good position tracking, with error margins well below 2 per cent. The force values were not estimated in the simulation but were formed according to the impedance law using the obtained tool position, velocity and acceleration values. Those expected force values are very close to the desired ones.

This paper presents a good starting point for the practical development of an impedance controller. Future work could include developing such a controller for commercial use, combining it with other control methods like admittance control, precisely estimating and measuring the controlled tool forces and other.

#### ACKNOWLEDGEMENT

The results presented in the paper are realized within the research project "Development and Experimental Performance Verification of Mobile Dual-Arms Robot for Collaborative Work with Humans", Science and Development Programme - Joint Funding of Development and Research Projects of the Republic of Serbia and the People's Republic of China, contract no. 401-00-00589/2018-9.

#### REFERENCES

- [1] N. Hogan, "Impedance Control: An Approach to Manipulation," *American Control Conference*, San Diego, CA, USA, 1984, pp. 304-313.
- [2] Andrews, J.R. and Hogan, N. (1983), "Impedance control as a framework for implementing obstacle avoidance in a manipulator", D. Hardt and W.J. Book (eds.), *Control of Manufacturing Processes and Robotic Systems*, ASME, New York, pp. 243-251
- [3] R. Ikeura and H. Inooka, "Variable impedance control of a robot for cooperation with a human," *Proceedings of IEEE International Conference on Robotics and Automation*, Nagoya, Japan, 1995, pp. 3097-3102 vol.3.
- [4] António Lopes and Fernando Almeida, "A force-impedance controlled industrial robot using an active robotic auxiliary device", *Robot. Comput. Integr. Manuf.* 24, 3 (June 2008), 299-309.
- [5] Al-Shuka, H., Leonhardt, S., Zhu, W. H., Song, R., Ding, C., & Li, Y. (2018). "Active Impedance Control of Bioinspired Motion Robotic Manipulators: An Overview", *Applied bionics and biomechanics*, 8203054
- [6] Almeida, Fernando & Lopes, António & Abreu, Paulo (1999), "Force-Impedance Control: a new control strategy of robotic manipulators.", *Recent advances in mechatronics..*
- [7] Bilal Komati, Cédric Clévy, Philippe Lutz, "Force Tracking Impedance Control with Unknown Environment at the Microscale", *IEEE International Conference on Robotics and Automation, ICRA'2014.*, Jan 2014, Hong Kong pp.1-6..
- [8] Chen, C., Nie, H., Chen, J., & Wang, X. (2014), "A velocity-based impedance control system for a low impact docking mechanism (LIDM)", *Sensors Basel, Switzerland*, (12), 22998-23016.

# Sistem za automatsku kompenzaciju sistemske greške senzora sile i momenta kod industrijskih robota

Nikola Vidaković, Nikola Knežević

*s r* —Ogroman je značaj merenja kontaktne sile u zadacima industrijske robotike. va merenja se mogu koristiti prilikom kontrole sile, zaštite od prekomernih sila, kao i u oblasti kolaboracije između čoveka i robota. Prilikom korišćenja senzora sile i momenta susrećemo se sa nekoliko problema koje ćemo u ovom radu pokušati da rešimo. Cilj nam je da razvijemo sistem za automatsku kompenzaciju sistemske greške senzora sile i momenta, koja će podrazumevati estimaciju pomeraja (engl. bias) senzora, gravitacionu kompenzaciju, kao i transformaciju sile u koordinatni sistem od interesa. Rezultat ove procedure je dobijanje preciznijih merenja kontaktnih sila koje su nam od značaja za dalju primenu.

**Ključne reči**—senzor sile i momenta gravitaciona kompenzacija estimacija mase i pomeraja automatska kompenzacija sistemske greške

## I. UVOD

Važnost preciznog merenja sila i momenata je esencijalna za mnoge primene u industrijskoj robotici kada robot dolazi u kontakt sa okolinom [1,2]. Merenja se mogu koristiti prilikom zaštite robota od prekomernih sila i kontrole primenjene sile [3,4]. U nekim slučajevima u industrijskoj robotici sama kontrola pozicije nije dovoljna, već je potrebno kontrolisati silu kojom se deluje [5,6]. Recimo, prilikom sklapanja određenih delova na traci za asembliranje je poznavanje sile neophodno [7,8,9]. Poznavanjem sile, greška u pozicioniranju gubi svoj značaj, jer se može lako detektovati kroz velike kontaktne sile, nakon čega možemo izvršiti korekciju pozicije [10]. Neki od novih izazova u polju robotike su kinematičko učenje [11] i kolaboracija između robota i čoveka [12,13], gde je poznavanje sile kojom čovek deluje na završni uređaj robota prilikom zajedničkog izvršavanja zadatka ključno.

U većini primena robot dolazi u kontakt sa okolinom preko svog završnog uređaja. Kontaktne sile koje deluju na završni uređaj mogu se meriti pomoću senzora koji se nalazi na zglobu pre završnog uređaja. Kasnije se može izvršiti transformacija ovih sila u odgovarajući koordinatni sistem, jer nam je često zgodnije da silu izražavamo u nekom fiksnom koordinatnom sistemu.

Usled različitih uticaja, kao što su promenljiva radna temperatura ili promene unutar elektronskih komponenti, može

se desiti da merenja senzora budu pomerena, tj. da ne pokazuju pravu vrednost sile, već vrednost koja ima određeni pomeraj. Takođe, uvek će postojati gravitaciona komponenta sile koja potiče od samog senzora i završnog uređaja koji je namontiran na njega, a koji imaju određenu masu. Da bismo dobili precizna merenja kontaktne sile, potrebno je izračunati konstantan pomeraj u merenjima senzora kao i promenljivu gravitacionu komponentu sile i kompenzovati ih prilikom očitavanja senzora.

Pošto se sa gorenavedenim problemima susrećemo prilikom svake primene senzora sile, značajno je osmisliti automatizovan sistem, koji će poništiti parazitne uticaje kada želimo da merimo kontaktnu silu. Taj sistem treba da objedinjuje sve neophodne procedure koje nam obezbeđuju precizna merenja. Takođe, procedura mora biti dovoljno modularna i univerzalna da se može koristiti kao sastavni deo različitih zadataka u robotici.

Ovaj rad daje odgovore na sledeća pitanja: Kako izvršiti estimaciju pomeraja senzora, kao i mase i centra mase senzora i završnog uređaja? Kako se vrši procena gravitacione komponente sile? Kako se vrši transformacija sile u različite koordinatne sisteme? Takođe, prikazani su rezultati eksperimentalnih merenja i diskusija o njihovoj preciznosti i mogućnostima primene razvijenih procedura.

Glavni cilj rada je razvoj automatizovane procedure za kompenzaciju sistemske greške senzora i dobijanje preciznih merenja kontaktne sile u globalnom koordinatnom sistemu.

## II. METODOLOGIJA RADA

Potrebno je osvrnuti se na teorijsku osnovu merenja sile kod industrijskih robota i postupke estimacije pomeraja, računanja gravitacione kompenzacije i transformacije sila u različitim koordinatnim sistemima. Ako sa  $s$  obeležimo vektor sa očitavanjima senzora, sa  $c$  vektor sa kontaktnim silama, sa  $b$  vektor sa pomerajima, sa  $g$  vektor sa promenljivim gravitacionim komponentama sila i sa  $n$  vektor sa očitavanjima usled šuma, onda možemo da pišemo:

$$\mathbf{F}_s = \mathbf{F}_c + \mathbf{F}_b + \mathbf{F}_g + \mathbf{F}_n, \quad (1)$$

gde važi da je  $\mathbf{F} = [F_x F_y F_z T_x T_y T_z]^T$ , pri čemu prve tri komponente predstavljaju projekciju vektora sile, a druge tri projekciju vektora momenta. Naš cilj je da izračunamo konstantan vektor sa pomerajima  $\mathbf{b}$  i promenljivi vektor  $\mathbf{g}$  tako da možemo što bolje da estimiramo vrednosti kontaktnih sila  $\mathbf{c}$ .

#### A. Estimacija pomeraja

Pomeraj je uključen u sva očitavanja senzora, tako da prvo moramo izračunati  $\mathbf{b}$  i onda oduzimati te vrednosti od očitavanja senzora tokom rada u realnom vremenu. Estimaciju vršimo iz niza merenja senzora koja ćemo očitavati u tačno određenim orijentacijama završnog uređaja. Merenja se očitavaju dok je robot u stanju mirovanja i neopterećen. Estimaciju pomeraja  $\mathbf{b}$  vršimo na osnovu sledeće ideje: recimo da računamo vrednost  $F_{bx}$ . Dovedemo robot u položaj u kom bi trebalo da  $F_x$  komponenta sile bude nula i u ovom slučaju je očitavanje senzora  $F_{sx}$  u stvari vrednost pomeraja  $F_{bx}$ . Alternativno, možemo dovesti robot u položaje u kojima bi sile trebalo da imaju vrednosti  $+F_x$  i  $-F_x$  i u ovom slučaju očitavanja će biti  $F_x + F_{bx}$  i  $-F_x + F_{bx}$  respektivno; tada pomeraj možemo naći kao aritmetičku sredinu ova dva očitavanja. Na sličan način može se estimirati i pomeraj momenata: tražimo pozicije u kojima su stvarne sile  $(+F_y, +F_z)$  i  $(-F_y, -F_z)$ , tada će očitavanja za moment oko x ose biti:

$$T_x = (r_y F_z - r_z F_y) + T_{bx}, \quad T_x = (-r_y F_z + r_z F_y) + T_{bx} \quad (2)$$

respektivno, gde je  $\mathbf{r}$  centar mase senzora. Vrednost  $T_{bx}$  možemo naći kao aritmetičku sredinu ova dva merenja. Kako bismo smanjili grešku estimacije usled šuma, možemo uzeti više od dva očitavanja sile za procenu pomeraja. Tačnije, postoje 24 odgovarajuće orijentacije senzora koje su pogodne za merenja pomoću kojih ćemo vršiti estimaciju [10]. To su pozicije u kojima bi dve komponente uvek trebalo da imaju vrednost nula, a treća  $\pm mg$ , gde je  $m$  masa senzora i završnog uređaja, a  $g$  lokalno gravitaciono ubrzanje. Pri tome se sve 24 pozicije mogu dobiti transformacijom globalnog koordinatnog sistema. Tada vrednosti pomeraja sile  $F_x$ ,  $F_y$  i  $F_z$  dobijamo kao aritmetičku sredinu svih 24 očitavanja odgovarajuće komponente. Vrednosti pomeraja momenta  $\tau_x$  dobijamo kao aritmetičku sredinu očitavanja u onim pozicijama koje odgovaraju gore opisanim slučajevima. Slično tome, estimiranu vrednost pomeraja  $T_y$  i  $T_z$  dobijamo usrednjavanjem očitavanja u njima odgovarajućim pozicijama.

#### B. Estimacija mase i centra mase

Da bismo izračunali gravitacionu komponentu sile, potrebno nam je da znamo masu i centar mase senzora i završnog uređaja. Ukoliko su ove vrednosti poznate mogu se koristiti, a u nastavku će biti dat postupak za njihovu estimaciju. Neka je gravitacioni vektor u globalnim koordinatama dat kao  $\mathbf{g}_1 = [0 \ 0 \ -g]^T$ . Ako je sa  $\mathbf{R}_i$  obeležena matrica rotacije između globalnog koordinatnog sistema i  $i$ -te pozicije senzora, odnosno završnog uređaja robota, tada će gravitacioni vektor izražen u koordinatnom sistemu senzora biti:  $\mathbf{g}_i = \mathbf{R}_i^T \mathbf{g}_1$ . Očitavanje senzora u  $i$ -toj poziciji nakon što je oduzet pomeraj iznosi  $\mathbf{c}_i =$

$m \cdot \mathbf{g}_i$ . Ako smestimo sva merenja sile od ranije (24 merenja po 3 sile) u vektor  $\mathbf{c}$  i sve vektore  $\mathbf{g}_i$  u vektor  $\mathbf{G}$ , možemo estimirati masu  $\bar{m}$  računajući:

$$\bar{m} = \min_m (\mathbf{F} - m\mathbf{G})^T (\mathbf{F} - m\mathbf{G}). \quad (3)$$

Koristeći metod najmanjih kvadrata [1], dobijamo:

$$\bar{m} = \frac{\mathbf{G}^T \mathbf{F}}{\mathbf{G}^T \mathbf{G}}. \quad (4)$$

Informacija o centru mase može se pronaći u merenjima momenta. Ako je rastojanje od centra senzorskog koordinatnog sistema do centra mase  $\mathbf{r}$ , tada ti momenat usled uticaja gravitacije, iznosi  $\boldsymbol{\tau}_i = m \mathbf{r} \times \mathbf{g}_{s1}$ . Vektorski proizvod iz ove relacije može se zameniti sledećim izrazom:  $\mathbf{T}_i = m \mathbf{A}_i \mathbf{r}$ , gde je

$$\mathbf{A}_i = \begin{bmatrix} 0 & g_{iz} & -g_{iy} \\ -g_{iz} & 0 & g_{ix} \\ g_{iy} & -g_{ix} & 0 \end{bmatrix}. \quad (5)$$

Sada smestimo sve momente u jedan vektor  $\mathbf{T}_{72 \times 1}$  i sve matrice  $\mathbf{A}_i$  u veliku matricu  $\mathbf{A}_{72 \times 3}$  i računamo vektor  $\bar{\mathbf{r}}$  kao:

$$\bar{\mathbf{r}} = \min_m (\mathbf{T} - \bar{m} \mathbf{A} \mathbf{r})^T (\mathbf{T} - \bar{m} \mathbf{A} \mathbf{r}), \quad (6)$$

gde je  $\bar{m}$  estimirana masa. Rešenje dobijamo korišćenjem pseudo-inverznog metoda:

$$\bar{\mathbf{r}} = \frac{1}{\bar{m}} (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{T}. \quad (7)$$

#### C. Gravitaciona komponenta sile

Kada znamo masu i centar mase, možemo izračunati gravitacionu komponentu sile:

$$\bar{\mathbf{F}}_g = \begin{bmatrix} \bar{m} \mathbf{R}^T \mathbf{g}_1 \\ \bar{m} \bar{\mathbf{r}} \times (\mathbf{R}^T \mathbf{g}_1) \end{bmatrix}. \quad (8)$$

Vidimo da je ovo promenljiva komponenta, jer zavisi od trenutne orijentacije senzora, odnosno od matrice rotacije  $\mathbf{R}$ , koja se može izraziti i preko Euler-ovih uglova. Odlučili smo se za ovakvu reprezentaciju matrice, jer Euler-ove uglove možemo dobiti od kontrolera robota putem direktne kinematike. Veza između elemenata matrice  $\mathbf{R}$  i Euler-ovih uglova glasi:

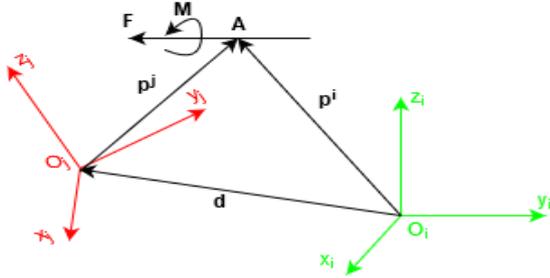
$$\mathbf{R} = \mathbf{R}_z(\varphi) \mathbf{R}_y(\theta) \mathbf{R}_x(\psi) = \begin{bmatrix} \cos(\varphi) & \sin(\varphi) & 0 \\ -\sin(\varphi) & \cos(\varphi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\psi) & -\sin(\psi) \\ 0 & \sin(\psi) & \cos(\psi) \end{bmatrix} = \begin{bmatrix} \cos(\theta) \cos(\varphi) & \sin(\psi) \sin(\theta) \cos(\varphi) - \cos(\psi) \sin(\varphi) & \cos(\psi) \sin(\theta) \cos(\varphi) + \sin(\psi) \sin(\varphi) \\ \cos(\theta) \sin(\varphi) & \sin(\psi) \sin(\theta) \sin(\varphi) + \cos(\psi) \cos(\varphi) & \cos(\psi) \sin(\theta) \sin(\varphi) - \sin(\psi) \cos(\varphi) \\ -\sin(\theta) & \sin(\psi) \cos(\theta) & \cos(\psi) \cos(\theta) \end{bmatrix}.$$

Konačno, kada zamenimo  $\mathbf{R}$  u gornji izraz dobijamo:

$$\bar{\mathbf{F}}_g = mg \begin{bmatrix} \sin(\theta) \\ -\cos(\theta) \sin(\Psi) \\ -\cos(\theta) \cos(\Psi) \\ r_z \cos(\theta) \sin(\Psi) - r_y \cos(\theta) \cos(\Psi) \\ r_x \cos(\theta) \cos(\Psi) + r_z \sin(\theta) \\ -r_y \sin(\theta) - r_x \cos(\theta) \sin(\Psi) \end{bmatrix}. \quad (9)$$

#### D. Transformacija sile i momenta u globalni koordinatni sistem

Često nam je od interesa da znamo kolika je vrednost kontaktne sile u nekom drugom koordinatnom sistemu, najčešće je to koordinatni sistem koji je pomeren na mesto vrha završnog uređaja (TCP – tool center point), a zadržava orijentaciju globalnog koordinatnog sistema. Time dobijamo merenje kontaktne sile na vrhu završnog uređaja izražene u fiksnim globalnim koordinatama.



Sl. 1. Transformacija sile i momenta između dva koordinatna sistema.

Transformacija se vrši na sledeći način:  $\mathbf{F}^i = \mathbf{T}_i^j \mathbf{F}^j$  (slika 1) gde je  $\mathbf{T}_i^j$  matrica 6x6 i možemo je napisati u sledećoj formi:

$$\mathbf{T}_i^j = \begin{bmatrix} \mathbf{R}_i^j & \mathbf{0} \\ \mathbf{W}_i^j \mathbf{R}_i^j & \mathbf{R}_i^j \end{bmatrix}, \quad (10)$$

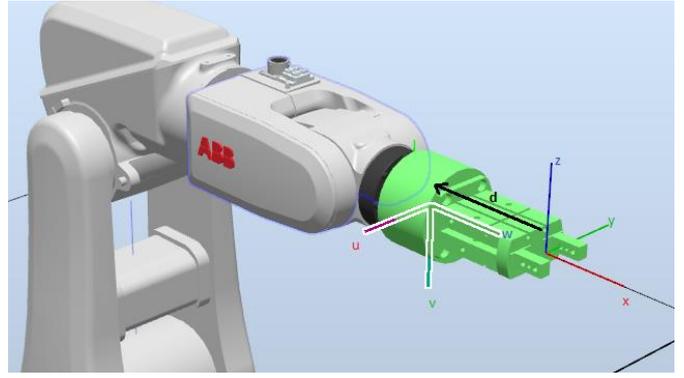
gde je

$$\mathbf{W}_i^j = \begin{bmatrix} 0 & -d_z & d_y \\ d_z & 0 & -d_x \\ -d_y & d_x & 0 \end{bmatrix}, \quad (11)$$

pri čemu je  $\mathbf{d}$  vektor pozicije  $O_j$  u odnosu na  $O_i$ , a  $\mathbf{R}_i^j$  matrica rotacije jednog koordinatnog sistema u odnosu na drugi [14]. Matricu  $\mathbf{R}_i^j$  možemo izračunati na način opisan u prethodnom poglavlju preko Euler-ovih uglova, a vektor  $\mathbf{d}$  je označava rastojanje između dva koordinatna sistema izražen u koordinatama globalnog koordinatnog sistema. Taj podatak takođe možemo dobiti od kontrolera robota, kao razliku pozicije dva koordinatna sistema u globalnim koordinatama. Konkretno, prilikom testiranja tačnosti sistema vršili smo merenja u jednoj konkretnoj poziciji (slika 2).

U našem slučaju vrednosti Euler-ovih uglova i vektora  $\mathbf{d}$  iznose:

$$\varphi = -90^\circ, \theta = 0^\circ, \Psi = -90^\circ, \\ d_x = -0.12 \text{ m}, d_y = 0 \text{ m}, d_z = 0 \text{ m}.$$



Sl. 2. Pozicija u kojoj smo vršili testiranje. Prikaz transformacije.

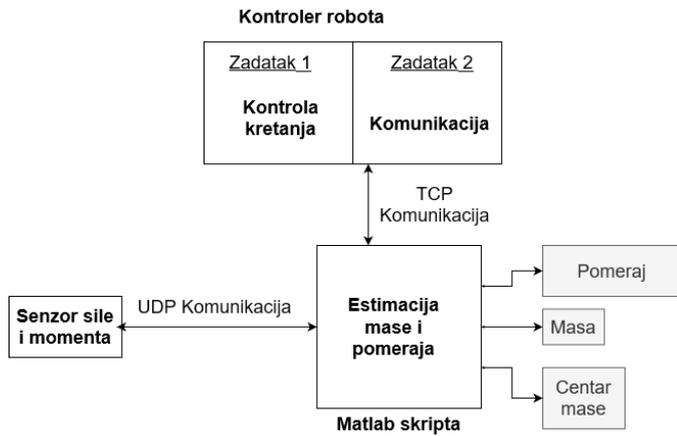
Odatve dobijamo rotacionu matricu:

$$\mathbf{R} = \begin{bmatrix} 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix}. \quad (12)$$

Na osnovu ovih vrednosti vršili smo transformaciju sile u vrh završnog uređaja sa globalnom orijentacijom. Ukoliko se robot kreće, podatke o orijentaciji senzora treba slati iznova kad god vršimo merenje sile, pošto će se ove vrednosti menjati, a menjaće se i komponente vektora  $\mathbf{d}$  u globalnom koordinatnom sistemu.

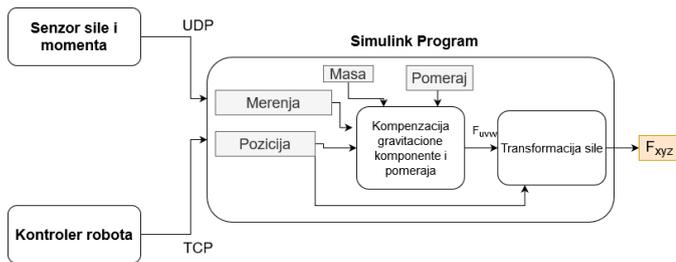
### III. IMPLEMENTACIJA

Ideja je da se pomoću Matlab(MathWorks,Inc.) programa na PC računaru, vrši komunikacija sa senzorom i kontrolerom robota, kao i obrada dobijenih podataka. Napisan je program u ABB RobotStudio okruženju, gde smo zadali kretanje robota, a komunikacija sa Matlab programom preko TCP protokola vrši se u paralelnom zadatku. Kada robot stigne u određenu poziciju, šalje se poruka matlab programu, koji tada vrši očitavanje merenja sa senzora. Komunikacija između Matlab programa i senzora realizovana je preko UDP protokola. Osim toga, u Matlabu su implementirani i algoritmi za estimaciju od ranije. Blok dijagram programa za estimaciju mase i pomeraja je prikazan na slici 3. Dakle, pokretanjem programa na kontroleru robota, a zatim i Matlab programa, izvršava se automatizovana procedura za računanje pomeraja, mase i centra mase i oni se čuvaju za dalju upotrebu. Takođe, realizovan je i Simulink blok koji ima zadatak da primi merenje sa senzora, kao i vrednost vektora  $\mathbf{d}$  i Euler-ovih uglova sa kontrolera robota i da izvrši sva potrebna izračunavanja i transformacije, kako bismo dobili željene kontaktne sile. Dakle, napisan je novi program za robot, gde su zadate odgovarajuće pozicije i putanje u kojima želimo da merimo kontaktnu silu, a u paralelnom task-u napisan je kod gde je izvršena TCP komunikacija sa Matlab programom, tako što se periodično šalje poruka koja sadrži trenutne vrednosti vektora  $\mathbf{d}$  i Euler-ovih uglova.



Sl. 3. Blok dijagram programa za estimaciju pomeraja i mase

Simulink blok komunicira sa senzorom i robotom i periodično dobija podatke o merenjima i poziciji, i izvršava sve potrebne transformacije i izračunavanja (računanje gravitacione komponente sile, i transformacija u globalni koordinatni sistem). Blok dijagram programa koji vrši komunikaciju sa robotom i senzorom i obradu merenja prikazan je na slici 4. Sva merenja se čuvaju na računaru i mogu se koristiti u daljoj analizi.



Sl. 4. Blok dijagram programa koji vrši komunikaciju sa robotom i senzorom i obradu merenja

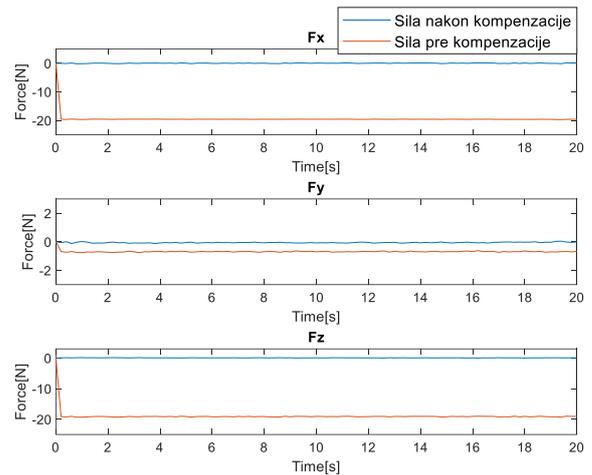
#### IV. REZULTATI

Eksperiment je sproveden u laboratoriji za robotiku Elektrotehničkog fakulteta, Univerziteta u Beogradu. Prilikom realizacije eksperimenata korišćen je ABB IRB 120 industrijski robot sa IRC5 kontrolerom i Ethernet Axia F/T Sensor [15]. Na robot je montiran senzor, a nakon njega i završni uređaj. Nakon pokretanja procedure za estimaciju pomeraja, mase i centra mase dobili smo sledeće rezultate:

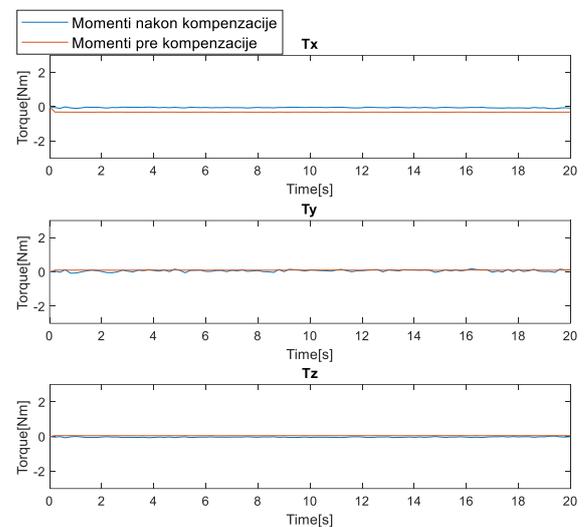
$$= [-19.77 \quad -6.88 \quad -19.36 \quad -0.07 \quad 0.09 \quad 0.07]^T, \quad m = 0.6271 \text{ kg},$$

$$\mathbf{r} = [0.0128 \quad -0.0001 \quad 0.0275]^T \text{ m}.$$

Zatim smo pokrenuli program koji vrši gravitacionu kompenzaciju i sve odgovarajuće transformacije, pri čemu nismo delovali nikakvom kontaktnom silom na završni uređaj robota. Na slikama 5 i 6 vidimo grafike kontaktnih sila nakon kompenzacije pomeraja i gravitacione komponente. Možemo zaključiti da smo dobili vrednosti bliske nuli.



Sl. 5. Sile „neopterećenog” robota u globalnom koordinatnom sistemu pre i posle kompenzacije

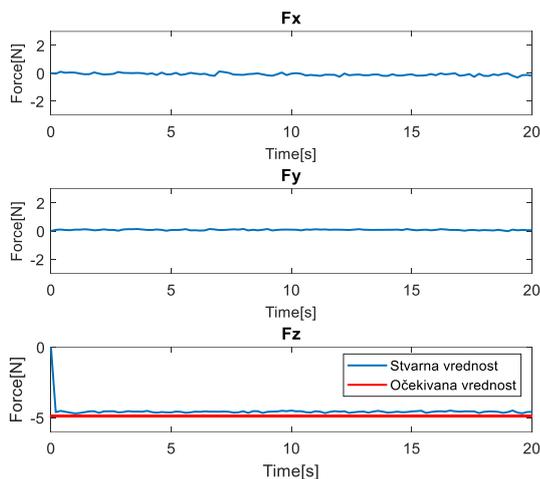


Sl. 6. Momenti „neopterećenog” robota u globalnom koordinatnom sistemu pre i posle kompenzacije

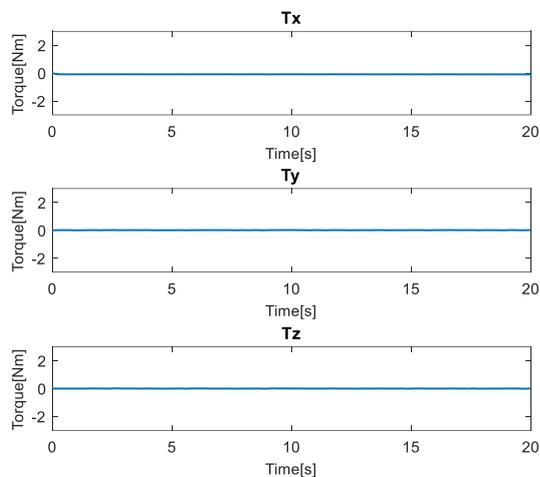
Kako bismo proverili preciznost senzora i uspešnost čitave procedure koristićemo teg od 500 grama koji ćemo staviti na završni uređaj (slike 7 i 8). Očekujemo da on doprinese očitavanju senzora kao  $F_z = -m \cdot g$  u globalnom koordinatnom sistemu, gde je  $m$  masa teža, a  $g$  gravitaciono ubrzanje i iznosi  $9.81 \text{ m/s}^2$ . Što se tiče momenta, pošto smo izvršili transformaciju sile u tačku u kojoj vršimo opterećenje, očekujemo nulte vrednosti svih momenata. Srednja vrednost sile  $F_z$  u ovom slučaju iznosi  $-4.5922 \text{ N}$ , a očekivali smo silu  $-4.905 \text{ N}$ . Greška iznosi  $0.3128 \text{ N}$ , odnosno  $6.38\%$ .

#### V. DISKUSIJA I ZAKLJUČCI

Na osnovu dobijenih rezultata možemo zaključiti da naša automatizovana procedura daje smislene rezultate. Na slikama 5 i 6 možemo videti da su vrednosti svih momenata i sila veoma bliske nuli, što znači da smo uspešno eliminišemo uticaj pomeraja senzora, kao i gravitacione komponente koja potiče od mase završnog uređaja i senzora.



Sl. 7. Sile sa tegom od 500 g na završnom uređaju robota



Sl. 8. Momenti sa tegom od 500 g na završnom uređaju robota

U nastavku eksperimenta, testirali smo tačnost naših merenja pomoću tega od 500g. Primećujemo da postoji greška 6.38%, koja potiče od nesavršenosti u estimaciji pomeraja i mase, ali i od greške u radu samog senzora. Uticaj šuma je primetan, pogotovu kod manjih sila. Iz specifikacija senzora i podataka o kalibraciji možemo zaključiti da ovaj model senzora i nije predviđen da radi sa malim silama, te se stoga i ne očekuje velika tačnost. Konkretno, za izabranu kalibraciju na y osi senzora koju smo koristili u eksperimentu, možemo meriti opseg od čak 200 N, pri čemu merna nesigurnost iznosi 2%, što je 4N. Dakle, treba proceniti koliki nam opseg merenja, kao i koja preciznost za konkretnu upotrebu treba, pa na osnovu toga izabrati senzor.

Treba napomenuti da bi estimaciju pomeraja trebalo raditi pre svakog korišćenja senzora, jer je primećeno da taj pomeraj nije uvek isti i da zavisi od različitih uticaja poput radne temperature, pritiska kojim smo pritegli završni završni uređaj na senzor i slično.

Procedure iznete u ovom radu predstavljaju osnov za dalju primenu senzora sile u industrijskoj robotici. Tek nakon što smo dobili tačnu vrednost kontaktne sile, možemo pristupiti njenoj kontroli ili zatvaranju povratne sprege. Jedna od mogućih primena je i detektovanje kontakta sa okruženjem

prilikom kretanja robota u prostoru, što se lako može demonstrirati. Veoma važna oblast gde se primenjuje senzor sile jeste i kolaboracija čoveka i robota, gde možemo meriti kojom silom čovek deluje na robota i u zavisnosti od nje sprovođiti određene akcije. Zatim, možemo sprovesti kinematsko učenje gde čovek deluje određenom silom u pravcu u kom želi da pomeri robota i time ga „učiči“ određenoj trajektoriji. Kao što vidimo, veoma je široka oblast primene merenja sile u robotici, pa samim tim i ovde razvijenih procedura. Puno je prostora i za dalje usavršavanje samih procedura u cilju minimizacije greške i dobijanju preciznijih merenja, kao i nadogradnji za neke kompleksnije primene.

Dakle, cilj ovog rada je razvoj automatizovane procedure za kompenzaciju sistemske greške senzora sile kod industrijskih robota. Ova procedura se koristi prilikom bilo kakvog zadatka gde se očekuje merenje kontaktnih sila. Osnovni koraci za primenu ove procedure u opštem slučaju su sledeći:

- Povezati svu potrebnu opremu na lokalnu mrežu (senzor, kontroler robota, PC računar) i pravilno montirati senzor i završni uređaj.
- Napisati program za očitavanje merenja sa senzora i komunikaciju sa robotom.
- Napisati program koji će dovesti robot u tačno određene pozicije, pri tome vodeći računa o komunikaciji sa programom za merenje.
- Nakon izvršenih merenja, potrebno je izvršiti estimaciju odgovarajućih parametara. U našem eksperimentu procedure su realizovane u Matlab-u i ABBRobotStudio okruženju, ali se mogu implementirati i na drugom softveru.
- Čuvamo izračunate parametre za dalji rad.
- U zavisnosti od primene, pišemo program za robot, imajući u vidu da je potrebno slanje pozicije robota u svakom trenutku u kom očekujemo merenje.
- Implementiramo program za očitavanje merenja, kompenzaciju pomeraja i gravitacione komponente, pomoću dobijenih parametara, a na gorenavedeni način. Ovaj deo se takođe može realizovati u Matlab-u ili nekom drugom programskom jeziku.
- Prethodno navedeni koraci čine našu automatizovanu proceduru za kompenzaciju sistemske greške, a rezultuju dobijanjem kontaktnih sila, koje koristimo u nastavku.

#### ZAHVALNICA

Ovaj rad je realizovan u okviru projekta TR 35003 koga finansira Ministarstvo prosvete, nauke i tehnološkog razvoja Republike Srbije, kao i od strane Fonda za nauku u okviru projekta PROMIS evidentiranog pod brojem 6062528.

#### LITERATURA

- [1] M. K. Vukobratović, V. Potkonjak, „Dynamics of contact tasks in robotics. Part I: general model of robot interacting with environment“, Mechanism and Machine Theory, Volume 34, Issue 6., pp 923-942, 1999
- [2] J. Jung, J. Lee, K. Huh. “Robust Contact Force Estimation for Robot Manipulators in Three-Dimensional Space.” Proceedings of the

- Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 220, no. 9, pp 1317–27, Sep. 2006.
- [3] L. Villani, J. De Schutter "Force Control". In: Siciliano B., Khatib O. (eds) Springer Handbook of Robotics. Springer, 2008
- [4] G. Zeng, H. Ahmad "An Overview of Robot Force Control." *Robotica*, vol. 15, no. 5 pp.473-82, Sep. 1997
- [5] R Hu, C. Long, L. Zhang „Satellite Assembly Technology by Robot Under Visual Guidance and Force Feedback Control”. Recent Developments in Mechatronics and Intelligent Robotics. *Advances in Intelligent Systems and Computing*, vol 856. Springer, 2018
- [6] W.D. Fisher, M. M. Shahid. "Hybrid Position/Force Control: A Correct Formulation." *The International Journal of Robotics Research* vol. 11, no. 4 pp. 299–311, Aug. 1992.
- [7] M. W. Abdullah, H. Roth, M. Weyrich, J. Wahrburg, „An Approach for Peg-in-Hole Assembling using Intuitive Search Algorithm based on Human Behavior and Carried by Sensors Guided Industrial Robot“, *IFAC-PapersOnLine*, Vol. 48, Issue 3, pp. 1476-1481, 2015.
- [8] Z. Jiabo, D. Na, D. Mingli, "Robotic Assembly Work Using Force Control," *IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC)*, Chongqing, China, pp. 1150-1153, 2018.
- [9] J. Bös, A. Wahrburg, K. D. Listmann, "Iteratively Learned and Temporally Scaled Force Control with application to robotic assembly in unstructured environments," *IEEE International Conference on Robotics and Automation (ICRA)*, Singapore, pp. 3000-3007, 2017.
- [10] S. Vougioukas, "Bias Estimation and Gravity Compensation for Force-Torque Sensors" *3rd WSEAS Symposium on Mathematical Methods and Computational Techniques in Electrical Engineering*, Athens, Greece, pp 82-85.
- [11] D. Massa, M. Callegari, C. Cristalli, "Manual guidance for industrial robot programming", *Industrial Robot*, Vol. 42 No. 5, pp. 457-465, 2015.
- [12] A. De Santis, B. Siciliano, A. De Luca, A. Bicchi, „An atlas of physical human–robot interaction“, *Mechanism and Machine Theory*, Volume 43, Issue 3, pp 253-270, 2008.
- [13] M. Geravand, F. Flacco, A. De Luca, "Human-robot physical interaction and collaboration using an industrial robot with a closed control architecture," *IEEE International Conference on Robotics and Automation*, Karlsruhe, pp. 4000-4007, May 2013.
- [14] B. Borovac, G. Đorđević, M. Rašić, M. Raković, "Industrijska robotika". Fakultet tehničkih nauka, Novi Sad, 2007.
- [15] *Ethernet Axia F/T Sensor Manual*, ATI Industrial Automation, Inc., Apex, North Carolina USA, 2019.
- [16] J. Craig, „Introduction to Robotics: Mechanics and Control“, Addison-Wesley, 1989

#### ABSTRACT

The measurement of contact force is very important in industrial robotics. These measurements can be used to control the applied force, protect the manipulator from excessive forces, as well as in the field of collaboration between humans and robots. When using force-torque sensors, we encounter several problems that we will try to solve in this paper. Our goal is to develop the automatic calibration system for force-torque sensors, which will include bias estimation, gravity compensation, and force transformation into a coordinate frame of interest. The result of this procedure is obtaining more precise measurements of contact forces, which are important for our further application.

#### **Automatic calibration system for force-torque sensors in industrial robotics**

Nikola Vidaković, Nikola Knežević

# Eye Gaze and Body Motion Synchronization in Dyadic Interaction

Danilo Nikić, Nikola Ilić, Darko Todorović, Nuno Ferreira Duarte, José Santor-Victor, Branislav Borovac, *Member, IEEE* and Mirko Raković, *Member, IEEE*

**Abstract** — Understanding the behavior alignment in dyadic human-human interaction and human-in-the-loop control in human-robot interaction relies on reliable tracking of the human motion. The gaze tracking and motion capture are common techniques that are used nowadays, but they are usually used separately. In this work we combined two Pupil-labs gaze tracker with a Vicon optical motion capture system. To synchronize the recordings of all devices we developed the solution that utilized Lab Streaming Layer for unified collection of measurement time series in research experiments that handles both the networking, time-synchronization and (near-) real-time access of the data. The aim of the experimental setup is to study the interaction of two humans while performing a joint task that requires interpersonal motion coupling.

**Index Terms**—human-human interaction, human-robot interaction, eye gaze, motion capture.

## I. INTRODUCTION

THE synchronized measurement of eye gaze, head gaze, trunk, and arm/hand movement enables insight into the action-perception process. This is especially important in dyadic interaction [1], for understanding interpersonal behavior coupling, mechanisms to read each other's intentions and anticipate the action of a co-worker. Such data quantifies the nonverbal communication that is comprehensive only when both (i) body parts motion capture and (ii) eye gaze tracking are merged and timely aligned.

Experimental setup (Fig. 1.) is prepared to gather the data for modeling intra- and interpersonal coupling and for developing a solution for action anticipation and consequently action planning in human-human (HHI) and human-robot interaction (HRI). The question of interest is whether the human is coupling eye gaze and body movements with his/her co-worker when picking the objects from the box, handing over the object, and placing them on the table. In HRI, eye gaze behavior, together with the activity of the hands and other parts of the body, has been argued to be important for anticipation and mutual alignment in the behavior [2-3].

D. Nikolić, N. Ilić, B. Borovac and M. Raković are with the Faculty of Technical Sciences, University of Novi Sad, Trg Dositeja Obradovića 6, 21000 Novi Sad, Serbia (e-mail: danilonikic4@gmail.com, nilic993@gmail.com, borovac@uns.ac.rs, rakovicm@uns.ac.rs).

D. Todorović is with the Faculty of Electronic Engineering, University of Niš, Serbia (e-mail: Darko.Todorovic@elfak.ni.ac.rs).

J. Santos-Victor, and N.F. Duarte are with the Institute for System and Robotics, Insituto Superior Tecnico, Av. Rovisco Pais 1, 1049-001 Lisboa, Portugal, University of Lisbon (e-mail: jasv@isr.tecnico.ulisboa.pt, tp.aobsilu.ocincet.rsi@etraudarierefn).



Fig. 1. Two human co-workers are picking the items from a box, scanning the barcode and handing over or placing the item on the table. During the experiment, eye gaze, head gaze, trunk, and arm motion is tracked with a Vicon motion capture system and Pupil-Lab gaze-tracking glasses.

Gaze tracking and motion capture are commonly used in many areas from psychology [4] and sports [5] to biomechanics [6] and robotics [7], but separately. The difficulty in fusing data from the different measuring system (i.e. motion capture system and gaze tracking), is time alignment and synchronization. Data acquisition and precise synchronization are especially significant for time-critical analysis of the data and for relating the different data streams to each other [8]. Interpersonal coupling, due to the nature of humans, is a stochastic process with asynchronous events, and precise time alignment is of paramount importance.

## II. STATE OF THE ART AND CONTRIBUTION

### A. Motion capture

Motion capture systems are available in different technologies [9]. Systems based on inertial measurement units are measuring the acceleration and angular velocity in three dimensions of a sensor attached to body parts. Another type of system is motion capture that is measuring position and orientation of the object in the magnetic field. In this research, we are using infrared camera-based system that is detecting the light reflected by markers attached to body parts and the objects. Optical motion capture system uses direct linear transformation to determine the exact position and orientation of cameras with respect to the reference coordinate frame. Then the minimum two cameras are needed to capture one marker in order to determine three-dimensional representation [6]. In order to track the position and orientation of the body, a minimum of three markers are needed to be attached to each body part. Different manufacturers are choosing between two

approaches for data labeling. Qualisys, for example, is recording the raw marker data and reconstructs the body motion after post-processing [9]. Manufacturers such as Vicon and OptiTrak allow the user to define the body model prior to the recording [10, 11]. The main challenge of optical systems is that occlusions of markers during the recording cause marker loss and gaps in the data. Thus, such occlusions should be prevented by careful marker placement and camera positioning before and during the recording.

### B. Eye gaze tracking

Most of existing gaze tracking systems can be divided into two groups, screen-based or head mounted gaze tracking glasses [12]. In this work we are using head mounted gaze tracker from Pupil-Labs [13]. It is a system of two cameras with an infrared light source for capturing eye movement and RGBD camera for recording the scene. Before the recording starts, it is necessary to calibrate the glasses, to match the actor's eyes to the model, for the estimation of the gaze point in the egocentric view. During the calibration, the actor needs to focus on the calibration marker on the screen while keeping the head still. Pupil-Labs gaze tracker can capture data at up to 200Hz and is able to estimate the position of the gaze in 3D.

### C. Synchronization of gaze tracking and motion capture

Synchronized gaze tracking and motion capture represent a multimodal information human behavior. Precise time alignment is important to extract relevant time events in the change of the behavior and its modeling. To synchronize multiple data streams, one can rely on sync boxes and plugins [14] or analog claps [15] equipped with motion capture markers captured by the gaze tracking glasses egocentric view camera. In [8] authors presented a method to synchronize Ergoneers Dikablis gaze-tracking glasses with Qualisys Oculus motion capture.

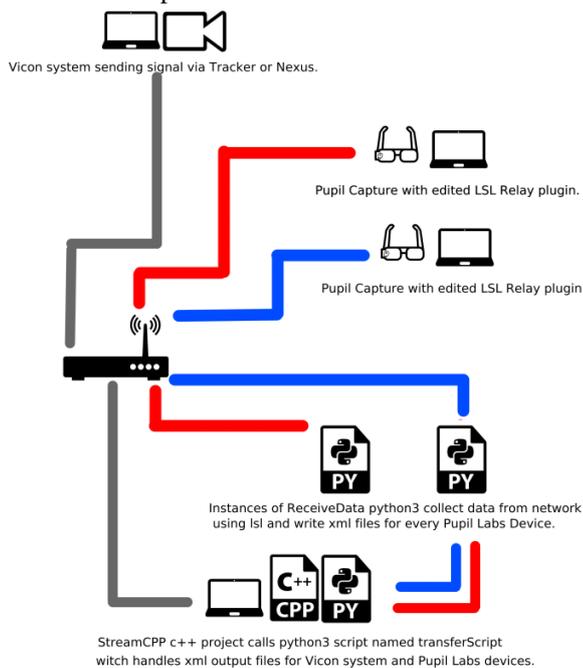


Fig. 2 Diagram of the experimental setup

In our work, as mentioned, we are syncing two Pupil-Labs gaze-tracking glasses with a Vicon motion capture system. The synchronization leverages on Network Time Protocol [16] and its implementation in lab-streaming-layer (LSL) [17]. The block diagram of the hardware and software components of our solution is given in Fig. 2.

Hardware-wise it is composed of Vicon cameras connected to Vicon Giganet MX box and dedicated PC, two Pupil-Labs glasses connected to laptops, a PC for the acquisition of data from different streams and a switch for establishing TCP/IP connection between all sensory systems. Software-wise, the system uses Vicon Nexus (Fig. 3.) for motion capture recording/streaming, Pupil capture for eye gaze recording/streaming, an application developed to stream data from Vicon system to LSL and application developed to capture and synchronize all data.

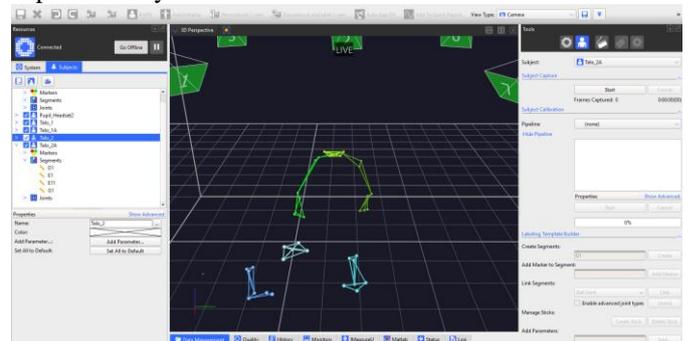


Fig. 3. Illustration of recording from Vicon Nexus while an actor (with markers placed on the glasses and both arms) interacting with a co-worker and manipulating the objects

First developed application (called StreamCPP) records the data on position and orientation of an object (rigid body) or subject (kinematic chain of rigid bodies) defined by markers in working environment whereas matching acquisition of raw data is done directly in Vicon Nexus. Subjects and objects in the workspace are recognized as defined instances. The data on their motion is streamed in real-time at 500Hz. Data acquisition is performed with the use of Vicon DataStream SDK (used version is 1.8.0). SDK can be used for the implementation of data streaming with different programming languages and operating systems. In this case, we used the C++ version of libraries for Windows operating system.

StreamCPP application establishes a connection with the PC that runs Vicon Nexus software through defined IP address. After the connection is successfully established, StreamCPP iterates through identified objects and subjects, enumerates them and extracts the information on the number of markers, and their assigned names. The main part of the application is a loop that is constantly updating the acquired frame (data package in specific time instance) streamed from Vicon Nexus. Acquired data are written in XML files („subject1.xml“, „subject.xml“, etc.). Another functionality of StreamCPP is to stream the acquired data to the LSL network for the purpose of synchronization with gaze tracking systems. Thus, for that purpose, StreamCPP creates LSL outlet and streams acquired data for each frame with an assigned time stamp from the Vicon system as well as LSL timestamp.

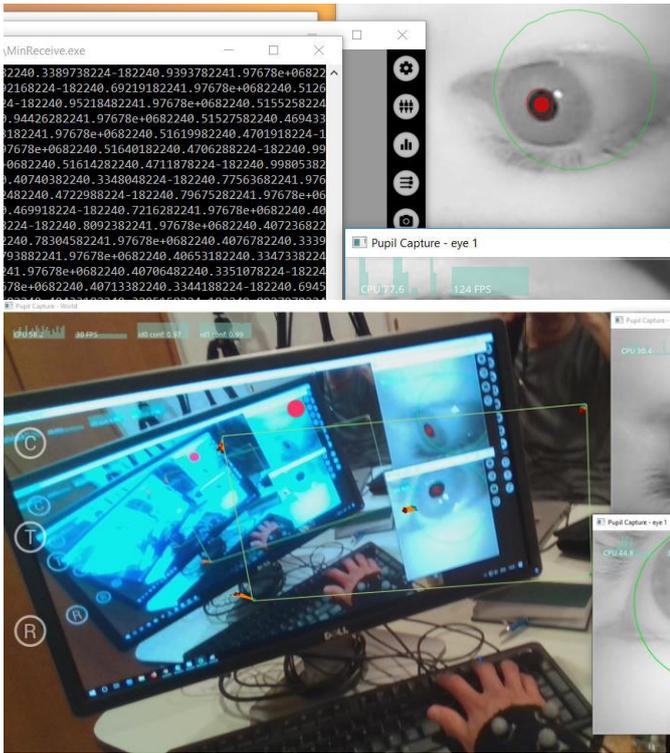


Fig. 4. Snapshots of the two eyes capture and egocentric view during data streaming

Gaze tracking and streaming are performed with Pupil Capture (Fig. 4). It captures eye movements with two infrared cameras and matches the eyeball, iris and eye gaze direction to the eyes model for estimation of gaze in 2D egocentric view camera and in 3D. Comparing to Vicon Nexus, Pupil Capture application does not provide data streaming directly with the installation. For this purpose, we used a plugin Pupil LSL Relay (pylsr) that enables streaming of the eye gaze data on LSL network. Pupil LSL Relay plugin opens outlet for each glass that is on LSL network. To acquire gaze-tracking glasses on LSL network, ReceiveData python script is developed. It writes the data to an XML file in the same format as in the case of StreamCPP application for motion capture data.

In order to synchronize all the streams, a Transferscript python script is developed. It is being called in each iteration of StreamCPP. Transferscript checks the total number of pupilOutput.xml files to determine the number of active gaze-tracking glasses. It also merges the timestamps of Vicon motion capture with Pupil-Labs glasses. That way, time synchronization of different sensing devices is performed<sup>1</sup>.

### III. INTERACTION EXPERIMENT SETUP

The purpose of this experimental setup is to prepare and conduct the human-human interaction experiment for evaluating the necessity of eye contact between pairs of subjects in a scenario which involves co-workers randomly picking objects from a box manipulating and placing or

<sup>1</sup> All the scripts and projects developed are accessible on the GitLab repository on the following link: <https://gitlab.com/vicon-pupil-data-parser>

handing over to a co-worker. An opaque blind is placed between co-workers with three different visibility conditions: first, the subjects can't see each other at all, next they can see up to shoulder height (head motion and eye gaze is occluded) and lastly the blind is removed completely. This is done with the intent of finding out how eye contact and/or arm motion visibility influence the subjects' choice of object picking, asynchronous request for involvement in interaction and frequency of collisions when picking the objects.

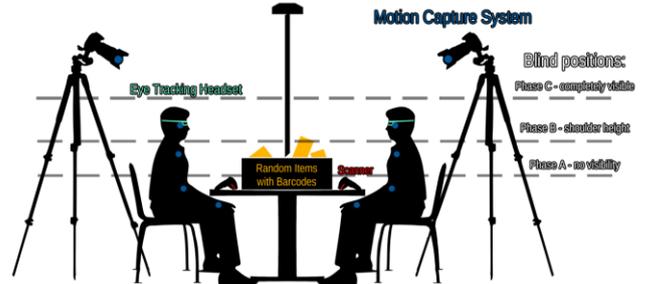


Fig. 5. Sketch of three different phases of the human-human interaction experiment

Pairs of subjects are seated facing each other, with a desk between them. On the desk is a box containing a number of items (product packaging with barcodes). Both subjects (simultaneously) pick out items from the box, scan the item's barcode with a scanner and then place it aside or handover it to a co-worker. The decision between placing and giving the item is made on the parity of bar code number (odd number is for placing and even for giving the item). The scanning in itself is significant as an additional step (action) for the subjects so that they have to focus on item manipulation and extract information on their task. The subjects are equipped with head-mounted eye trackers and surrounded by a motion tracking camera system which tracks markers placed on their arms and on eye tracking glasses. The box contains 24 items, 12 with odd and 12 with even bar code numbers. The experiment involves a total of 18 people divided into pairs. Each pair will be involved in one out of three visibility conditions of the experiment.

In our previous work [2, 3] we used a similar setup to study the importance of eye gaze for understanding and anticipating the interaction and to develop gaze behavior model to control the eye gaze and arm movements of the robot. The experiment involved questioner given to human subjects to anticipate the action that the actor will perform based on different available cues. Here we are directly disturbing the interaction with the cover placed in between the co-workers and we will be able to better understand the role of different non-verbal cues in human-human interaction and transfer that knowledge to the models for controlling humanoid robot co-worker.

#### A. Procedure for experiment data acquisition

At the beginning of the experiment, two subjects are first equipped with gaze tracking glasses and markers placed on their arms. Before each data acquisition, first, it is necessary to calibrate all the systems, i.e. motion capture, as well as two gaze-tracking glasses. Next step requires the creation of the subjects to match the body parts of the experiment

participants. Vicon Nexus and Pupil Capture have to be validated if they are recognizing the subjects and are properly estimating the gaze. In case of too many false measurements, the calibration procedure should be repeated. Once the motion and gaze tracking are reliable, the internal recording (in Vicon Nexus and Pupil Capture) can start. After starting the internal data recording, all the scripts and application for LSL data streaming and synchronization has to be started. It is necessary to check if all the devices are sending streams on LSL network. When all the data are streamed and recorded properly, the experiment can start. Once the experiment is finished, all applications and scripts have to be stopped, and it is necessary to check if all the output files are properly saved and closed.

#### IV. CONCLUSION

In this work, we presented the experimental setup for acquiring a multimodal dataset containing eye gaze and body motion during human-human interaction. The setup involves a Vicon Motion tracking system and two Pupil-Labs gaze-tracking glasses. The acquired data are saved in raw format in manufacturer's recording software. For synchronizing the data obtained from different measuring equipment we developed applications and scripts that are streaming the data to LSL network and capture that data at the same place where timestamps of different data sources are matched. Our next steps will involve preparation of a dataset for the experimental setup that is described in Section III. that includes annotating specific events and fixations during interaction so that temporal correlations between important events can be modeled. Further on we will focus on improving the models for upper body humanoid robot control based on the derived model.

#### ACKNOWLEDGMENT

This work was partially supported by MNTR project III44008 and TR35003, EU H2020 project ACTICIPATE and

FCT project RBCog-Lab research infrastructure.

#### REFERENCES

- [1] M. Gallotti, M.T. Fairhurst, C.D. Frith, "Alignment in social interactions", *Consciousness and cognition*, 48, pp. 253-261., 2017
- [2] N. F. Duarte, M. Raković, J. Marques, J. Santos-Victor, "Action Alignment from Gaze Cues in Human-Human and Human-Robot Interaction" In Proceedings of the European Conference on Computer Vision (ECCV), 2018.
- [3] N. F. Duarte, M. Raković, J. Tasevski, M.I. Coco, A. Billard, J. Santos-Victor, "Action anticipation: Reading the intentions of humans and robots", *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4132–4139, Oct. 2018. <https://doi.org/10.1109/LRA.2018.2861569>
- [4] D. Preissmann, C. Charbonnier, S. Chagué, J.P. Antonietti, J. Llobera, F. Ansermet, P. J. Magistretti, "A motion capture study to measure the feeling of synchrony in romantic couples and in professional musicians.", *Frontiers in psychology*, vol. 7, 2016.
- [5] E. van der Kruk, M. Reijne, "Accuracy of human motion capture systems for sport applications; state-of-the-art review.", *European journal of sport science*, nol.18, no. 6, pp. 806-819, 2018.
- [6] G.E. Robertson, G.E. Caldwell, J. Hamill, G. Kamen, S. Whittlesey, "Research methods in biomechanics", Human kinetics, 2018
- [7] C. Mandery, Ö. Terlemez, M. Do, N. Vahrenkamp, T. Asfour, "The KIT whole-body human motion database", In *2015 IEEE International Conference on Advanced Robotics (ICAR)*, pp. 329-336, 2015
- [8] B. Burger, A. Puupponen, T. Jantunen, "Synchronizing eye tracking and optical motion capture: How to bring them together.", *Journal of Eye Movement Research*, vol. 11, no. 2, 2018.
- [9] Qualisys. <http://www.qualisys.com>
- [10] Vicon. <http://www.vicon.com>.
- [11] Optitrack, <https://optitrack.com/>
- [12] A. Duchowski, "Eye tracking methodology", Springer, 2007.
- [13] M. Kassner, W. Patera, A. Bulling, "Pupil: an open source platform for pervasive eye tracking and mobile gaze-based interaction", In: Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing: Adjunct publication, ACM, pp. 1151–1160, 2014
- [14] L. Bishop, W. Goebel, "Mapping visual attention of ensemble musicians during performance of "temporally-ambiguous" music." In *Conference of Music & Eye-Tracking*. 2017.
- [15] F. Marandola, "Eye-Hand synchronization and interpersonal interaction in xylophone performance: A comparison between African and Western percussionists", *Journal of Eye Movement Research*, vol.11, no.2, 2019,
- [16] D.L. Mills, "Internet time synchronization: the network time protocol", *IEEE Tran. on communications*, Vol. 39, no. 10., pp 1482-1493, 1991
- [17] C. Kothe, "Lab streaming layer (LSL)", Accessed on April 26., 2019., <https://github.com/scn/labstreaminglayer>

# Biologically Inspired Optimization Methods for Image Registration in Visual Servoing of a Mobile Robot

Lazar Đokić, Aleksandar Jokić, Milica Petrović and Zoran Miljković

**Abstract**—Image registration (IR) represents image processing technique that is suitable for use in visual servoing (VS). This paper proposes the use of Biologically Inspired Optimization (BIO) methods for IR in case of nonholonomic mobile robot. The comparison study of three different BIO methods is conducted, namely Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Grey Wolf Optimizer (GWO). The aforementioned optimization algorithms utilized for IR are tested on images of manufacturing entities acquired by mobile robot stereo vision system. The considered algorithms are implemented in the MATLAB environment. The experimental results suggest satisfactory geometrical alignment after IR, whilst GA and PSO outperform

**Index Terms**—Image Registration, nonholonomic mobile Robot visual servoing, Biologically Inspired Optimization

## I. INTRODUCTION

A vision-based control, also known as Visual Servoing (VS), is referred to the use of computer vision data for motion control of intelligent robotic systems [1]. An intelligent Mobile Robot (MR), equipped with two cameras, by its own movement induces camera motion. Therefore, a MR can directly control its movement based on the information obtained from images acquired by cameras. VS involves continuous measurement of the visually observed error between target and current image. MR utilizes computer vision to create a feedback signal and produce the required movement until the aforementioned error reaches zero or

defined error threshold. Continuous measurement and feedback control provide great robustness to errors in the system [2]. As is clear, VS heavily relies on image processing for the extraction of useful information such as geometric feature extraction (points, lines, edges, etc.), object classification, pattern recognition, etc. In order to mitigate computationally demanding and time-consuming methods for feature extraction, direct methods that exploit pixel intensities can be used [3]. VS approaches that do not require metric information of the object, its shape, or camera motion produce more versatility and robustness to the error. Direct VS implies the use of pure image signal to design the vision-based control law and the function of Image Registration (IR) is to recover unknown parameters directly from measurable image quantities at each pixel in the image [4]. Accordingly, intensity-based IR can be used to construct control error from the projective parameters that geometrically relate the current image with the target one [5]. In both VS and IR, one of the images is referred to as target (fixed) image, and the other one is the current (moving) image. This similarity allows for the straightforward implementation of IR techniques in VS as proposed in [3] and [5]. In the previously mentioned papers, VS that utilizes IR for image processing is referred to as direct VS. IR, in general, represents the process of geometrical alignment of two images (the target and current image) and can represent a key step in image preparation for the seamless execution of VS. Several different techniques of IR are proposed in the literature, and we address papers where Biologically Inspired Optimization (BIO) methods are used. IR is most commonly used for medical imagery, and it comes as no surprise that BIO is mostly used in medical IR for combining computer tomography and nuclear magnetic resonance data [6]. The authors of [7] have done a comprehensive review of Particle Swarm Optimization (PSO) algorithm in multimodal medical IR and some of the successful applications are presented in [8] and [9]. They concluded that intensity-based IR usually requires the optimization of some similarity metric, and global optimization methods, such as PSO, represent very efficient and effective methods that achieve encouraging results in medical IR. Besides PSO, Genetic Algorithm (GA) still finds application in a number of problems. The authors of [10] employed GA optimization for choosing the optimal values of IR parameters. Similarly, the paper [11] proposed multimodal intensity-based IR based on real-coded GA. The

Lazar Đokić, PhD student, University of Belgrade - Faculty of Mechanical Engineering, Department of Production Engineering, Laboratory for industrial robotics and artificial intelligence (ROBOTICS&AI), Kraljice Marije 16, 11120 Belgrade 35, The Republic of Serbia (lazar.m.djokic@gmail.com).

Aleksandar Jokić, PhD student, University of Belgrade - Faculty of Mechanical Engineering, Department of Production Engineering, Laboratory for industrial robotics and artificial intelligence (ROBOTICS&AI), Kraljice Marije 16, 11120 Belgrade 35, The Republic of Serbia (ajokic@mas.bg.ac.rs).

Dr. Milica Petrović, Assistant Professor, University of Belgrade - Faculty of Mechanical Engineering, Department of Production Engineering, Laboratory for industrial robotics and artificial intelligence (ROBOTICS&AI), Kraljice Marije 16, 11120 Belgrade 35, The Republic of Serbia (mmpetrovic@mas.bg.ac.rs).

Dr. Zoran Miljković, Full Professor, University of Belgrade - Faculty of Mechanical Engineering, Department of Production Engineering, Laboratory for industrial robotics and artificial intelligence (ROBOTICS&AI), Kraljice Marije 16, 11120 Belgrade 35, The Republic of Serbia (zmiljkovic@mas.bg.ac.rs).

aforementioned BIO methods showed improved convergence speed compared to the traditional IR methods, with a more relevant exploration of the search solution space and better alignment accuracy. Recently, the Grey Wolf Optimization (GWO) has emerged as one of the best-performed metaheuristic BIO algorithms for the vast amount of engineering applications [12].

Therefore, in this work, the authors analyze the three above mentioned BIO algorithms for IR application. Moreover, another major difference compared to the previously mentioned VS approaches is the implementation of BIO methods for intensity-based IR in VS. Similar to direct VS, whole images obtained at different viewpoints during the motion of intelligent MR are analyzed. Images are acquired by using two camera sensors (acA1920-25uc - Basler ace area scan cameras). Utilized images are obtained by the stereo vision system of nonholonomic MR RAICO (Robot with Artificial Intelligence based COgnition) in the laboratory model of the manufacturing environment (Fig. 1).

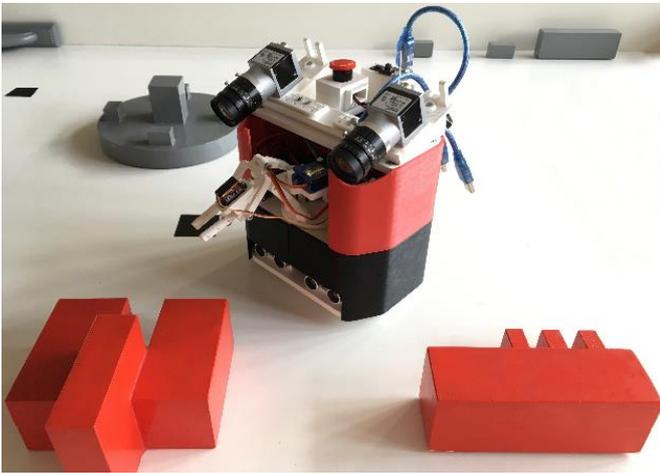


Fig. 1. Nonholonomic mobile robot RAICO

The paper structure is as follows. In Section II problem of intensity-based IR is formulated. Section III contains a brief description of different BIO algorithms while in Section IV Fitness Function (FF) is defined. Comparison of BIO algorithms for intensity-based IR is presented in Section V and Section VI gives concluding remarks.

## II. INTENSITY-BASED IMAGE REGISTRATION

The function of intensity-based IR in VS is to geometrically relate two images taken from various MR poses (i.e. position and orientation). Target images are obtained by positioning MR in the desired pose, and current images are taken at different camera viewpoints. In order to successfully carry out intensity-based IR, the current image must contain the same manufacturing entity that is on the target image. Prior to intensity-based IR, acquired images are preprocessed, and afterward, two binary images of manufacturing entities are geometrically aligned. Fig. 2 show the target images of manufacturing entities before and after preprocessing.

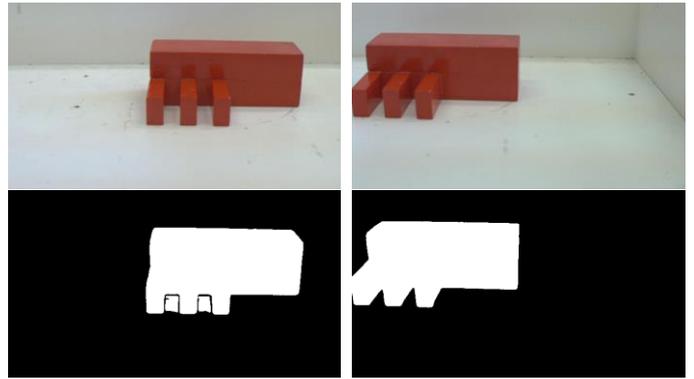


Fig. 2. The target images before and after preprocessing

It must be noted that target images are binarized before motion initialization of MR and binarization of current images is done while moving. Although the preprocessing is not computationally demanding, in combination with BIO real-time implementation is still not possible. A similar vision-based control strategy as proposed in [13] can be implemented for motion control, where the appropriate moving sequence depends on the ratio of translational velocities. The goal of intensity-based IR is to determine a spatial Transformation Matrix (TM) that matches two images:

$$T = \begin{bmatrix} s \cdot \cos(\theta) & -s \cdot \sin(\theta) & t \\ s \cdot \sin(\theta) & s \cdot \cos(\theta) & t \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

where  $s$  and  $s$  are scaling parameters with respect to  $x$  and  $y$  axis,  $\theta$  is angle of image rotation, and  $t$  and  $t$  represent translation along  $x$  and  $y$  axis, respectively.

Dimension of TM is 3x3 and via BIO, optimal values of TM elements for geometrical alignment of two rigid bodies (manufacturing entities) are acquired.

## III. BIOLOGICALLY INSPIRED OPTIMIZATION METHODS

Many optimization problems can be solved using biologically inspired stochastic optimization algorithms. BIO methods present computationally efficient alternatives to deterministic methods. These BIO methods are population-based approaches that start from a randomly initialized population of solutions and iteratively improve them during the optimization process. Individual solutions with the best fitness function values are kept thorough iterations and new individuals are generated until the maximum number of iterations is reached. At the end of all iterations, the best solution is considered an optimal one. This represents the main disadvantage of BIO methods because there is no guarantee that the found solution is actually the optimal one [14]. There are also some cases where BIO algorithm can be stopped due to the convergence to the local extremum before all iterations are done, defined with early stopping criterium. BIO methods have clear advantages compared to traditional deterministic optimization methods in solution exploration and their wide application in solving various problems is

considerable.

A brief description of the analyzed BIO methods for IR in VS are given below.

#### A. Genetic Algorithm

GA belongs to the class of evolutionary algorithms inspired by natural evolution. The generated population consists of chromosomes (individuals) which are modified with GA operators (crossover and mutation) in order to converge to the solution with the best value of FF. The main steps of GA are (i) initialization of GA parameters; (ii) generation of individuals for an initial population and FF evaluation of the individuals; (iii) selection; (iv) crossover; and (v) mutation. The process is usually repeated until the desired number of generations is reached. GA can be considered slow for finding an optimal solution but it can explore complex space and find values of FF close to the global optimum [15].

#### B. Particle Swarm Optimization

PSO is considered a biologically based technique of artificial intelligence, inspired by the collective intelligence of swarm (e.g. bird flock, fish school, etc.). The generated population (swarm) consists of individuals (particles) that are adjusting their velocity (2) accordingly to their currently best solution ( $P_{id}^t$ ) and based on information obtained in interaction with other individuals ( $P_{gd}^t$ ). The whole swarm shares the best position of a single particle, as given by (3):

$$v_{id}^{t+1} = v_{id}^t + C_1 \cdot r \cdot (P_{id}^t - v_{id}^t) + C_2 \cdot r \cdot (P_{gd}^t - v_{id}^t) \quad (2)$$

$$p_{id}^{t+1} = v_{id}^t \cdot t \quad (3)$$

where  $t$  represents the current iteration,  $p_{id}$  is the particle position,  $r$  is a random number [0,1],  $C_1$  and  $C_2$  are acceleration constants, while  $w$  denotes the inertia parameter. PSO represents an efficient global optimization algorithm with fast convergence speed that can be easily implemented. One of the known shortcomings of PSO algorithm is the probability of convergence to the local optimal solution in the early stages of optimization.

#### C. Grey Wolf Optimization

Grey Wolf Optimizer (GWO) is a metaheuristic algorithm inspired by behavior of grey wolves. This algorithm mimics the leadership hierarchy and hunting mechanism of grey wolves in nature [12]. The societal hierarchy of the grey wolf pack is divided into four groups (alpha -  $\alpha$ , beta -  $\beta$ , delta -  $\delta$ , and omega). The distance between wolves in the pack is calculated with (4):

$$D_a = |p_{i\alpha}(t) - p(t)|, \quad a = \alpha, \beta, \delta; \quad i = 1, 2, 3, \quad (4)$$

where  $D$  represents the distance,  $p(t)$  is a wolf position in the  $t$ -th integration and  $r$  represents a random number in [0, 2]. Every group has its role in the pack and according to its group status the best FF is defined by alpha, then beta and delta, and the rest of the solutions are considered to be omega. GWO is represented by pack hunting and three main steps are

searching for prey; encircling prey; and attacking prey. Position of alpha, beta, and delta leaders are given by (5), while new position of each wolf is updated by (6):

$$p_1 = p_\alpha - A_1(D_\alpha), \quad p_2 = p_\beta - A_2(D_\beta), \quad p_3 = p_\delta - A_3(D_\delta) \quad (5)$$

$$p(t+1) = (p_1 + p_2 + p_3) / 3 \quad (6)$$

The GWO saves the best solutions obtained through a defined number of iterations and the goal is to reach prey by the shortest possible route [17].

#### IV. FITNESS FUNCTION

In this paper, Fitness Function (FF) provides an assessment of the geometrical alignment of the target and current image. The lower value of FF implies better geometrical alignment, so the goal of BIO is to minimize proposed FF. For evaluating of FF, Sum of Squared Differences (SSD) for two images is used (7):

$$SSD(I_1, I_2) = \sum_{(u,v) \in I} (I_1[u,v] - I_2[u,v])^2, \quad (7)$$

where  $I_1$  and  $I_2$  are the target and current image spatially transformed by TM to match the target image, and  $u$  and  $v$  are pixel coordinates of given images.

Solution with a minimal value of FF is considered to be an optimal solution. Therefore, appropriate velocities of MR can be computed based on optimally generated elements of TM.

#### V. EXPERIMENTAL RESULTS COMPARISON

In this section, the implementation of BIO algorithms (described in Section III) for IR in VS is analyzed. Images used for IR are taken by positioning nonholonomic MR RAICO in poses with known pose displacements given in Table I. RAICO coordinate system is defined according to Fig. 3. A total of 24 images (12 pairs of stereo images) with different displacement from the target images are used for testing of IR.

TABLE I  
CURRENT IMAGE DISPLACEMENT COMPARED TO THE TARGET IMAGE

air of img.	pose cm, cm, °			air of img.	pose cm, cm, °		
	x	z	$\theta$		x	z	$\theta$
#1	0	+2	0	#7	+2	0	0
#2	0	-2	0	#8	+4	0	0
#3	0	-4	0	#9	+2	-2	0
#4	0	-6	0	#10	+4	-4	0
#5	-4	0	0	#11	0	0	+5
#6	-2	0	0	#12	0	0	-5

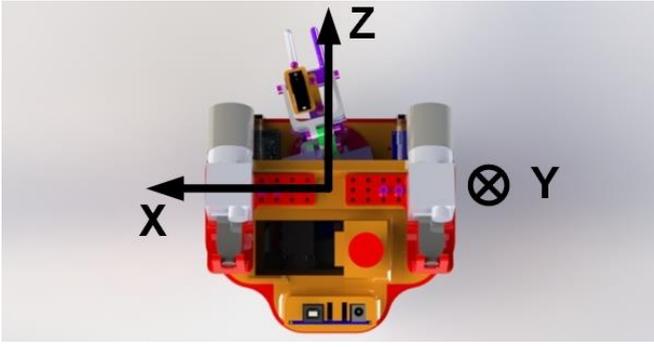


Fig. 3. Coordinate system of nonholonomic mobile robot RAICO

The selection of optimal optimization parameters is done through preliminary experimental parameters tuning. Optimization parameters that are the same for all three BIO algorithms (GA, PSO, and GWO) are: (i) the size of the population – 50, (ii) the maximum number of iterations – 50, and (iv) the number of design variables – 5. For GA, mutation probability is determined by Gaussian distribution, and crossover probability is 0.8. PSO acceleration constants ( $C_1$  and  $C_2$ ) are set to 2, and the inertia parameter is set to adaptive, with minimum and maximum value of 0.1 and 1.1, respectively. Evaluated design variables are elements of TM ( $\theta, s, s, t, \text{ and } t$ ), and their lower and upper bounds are defined as follows:

$$b = [-5, 0.7, 0.7, -20, -20] \quad (8)$$

$$Ub = [+5, 1.2, 1.2, 20, 20] \quad (9)$$

where Lb is lower bound for  $\theta, s, s, t, \text{ and } t$ , respectively, and Ub is upper bound for  $\theta, s, s, t, \text{ and } t$ , respectively.

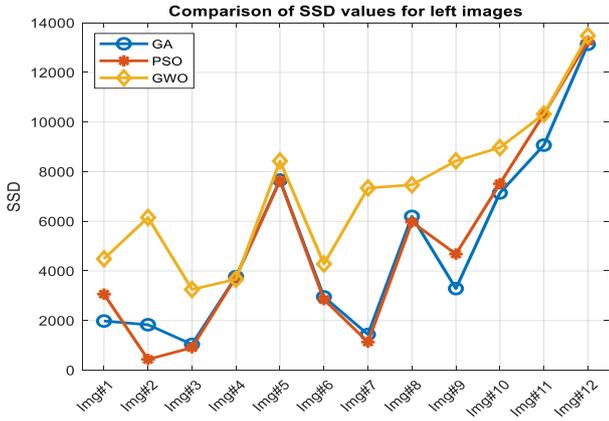


Fig. 4. Comparison of SSD values for left images

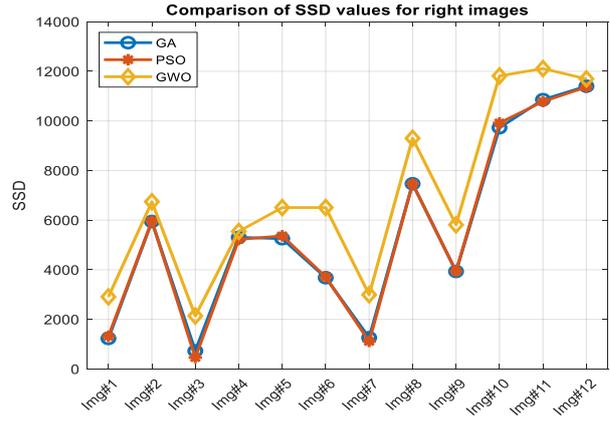


Fig. 5. Comparison of SSD values for right images

Comparison of Sum of Square Differences (SSD) values for left and right images of stereo pairs are given in Fig. 4 and Fig. 5, respectively. It can be observed that with the increase of disparities between the target and current image, the value of SSD steadily increases. Also, for the corresponding left and right stereo pair images, different values of SSD are calculated. In some cases, this occurrence can result in computing different MR velocities for the same camera viewpoints. As it can be seen from Fig. 4 and Fig. 5 the best optimization results are obtained by GA and PSO algorithm, while the worst results in this study are procured by GWO algorithm.

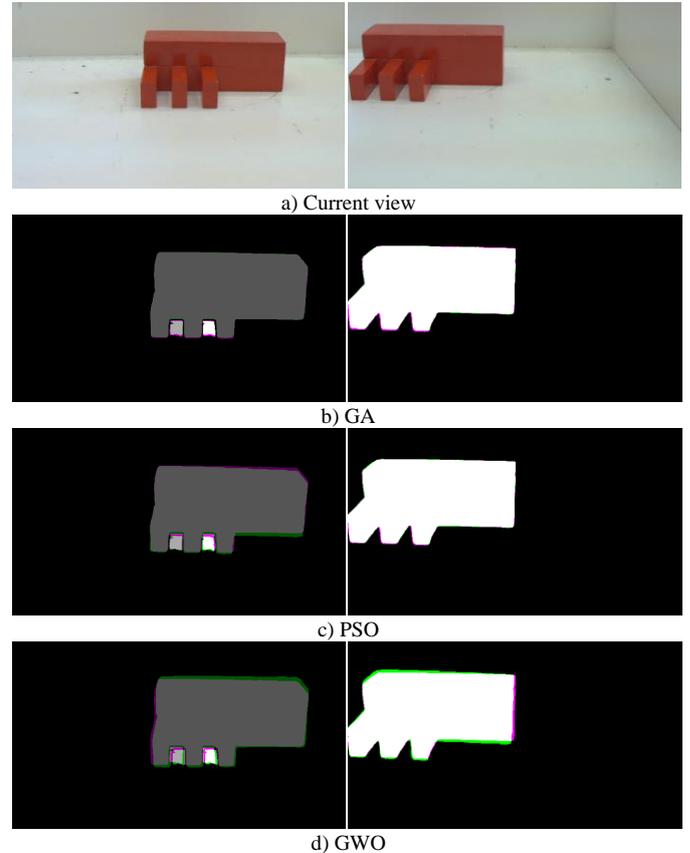


Fig. 6. Comparison of BIO methods for IR – Img#3

A comparison of geometrical alignment after IR for images #3, images #7, and images #12 with target images (Fig. 2) are given in Fig. 6, Fig. 7, and Fig. 8, respectively. When FF of BIO methods is minimized, elements of TM are evaluated in such a way that the current image can be spatially transformed to overlap with the target image. In Fig. 6 and 7, the overlapping of current and target image is obvious when GA and PSO are utilized. The assessment of satisfactory geometrical alignment is grey and white overlapping, while green and pink colors suggest an unsatisfactory geometrical alignment of current and target image. Fig. 7 shows overall poor geometrical alignment due to inadequate evaluation of TM elements. For both left and right current images, IR for image #10, image #11, and image #12 cannot be considered successful. Large initial displacements have negatively affected the convergence of BIO methods to the global optimal solution. It also should be noted that initial rotation ( $-5^\circ$ ) about the y-axis in image #12 has caused the manufacturing entity to be only partially seen (Fig. 8a)).

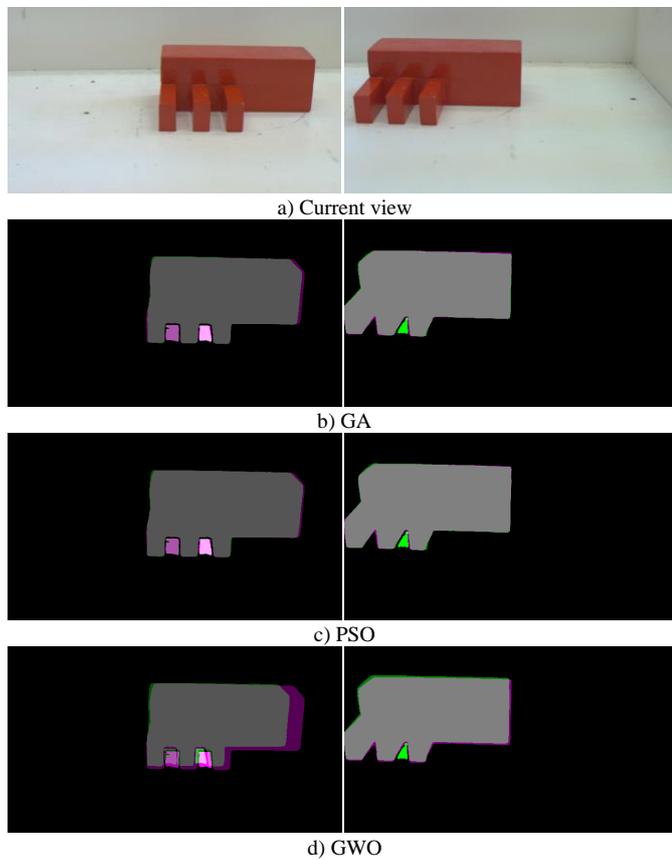
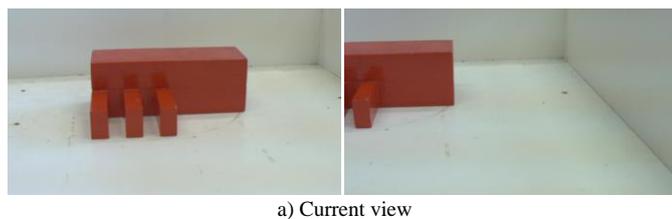


Fig. 7. Comparison of BIO methods for IR – Img#7



a) Current view

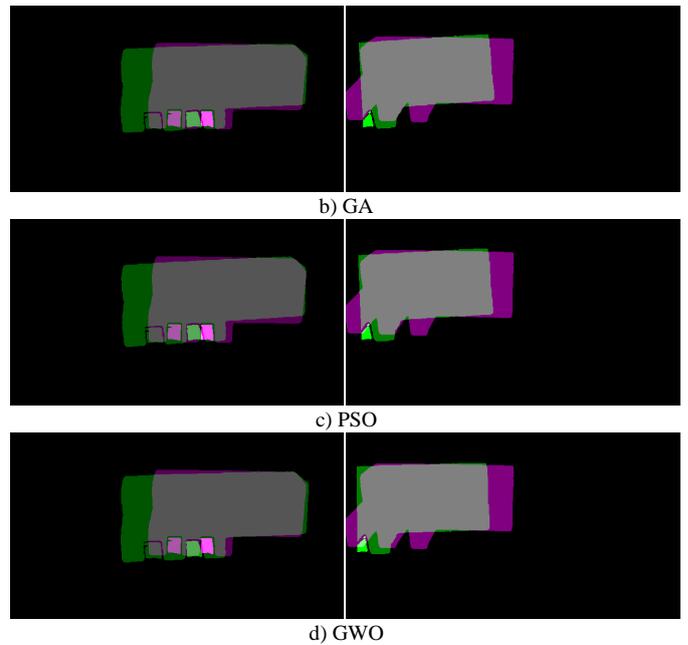


Fig. 8. Comparison of BIO methods for IR – Img#12

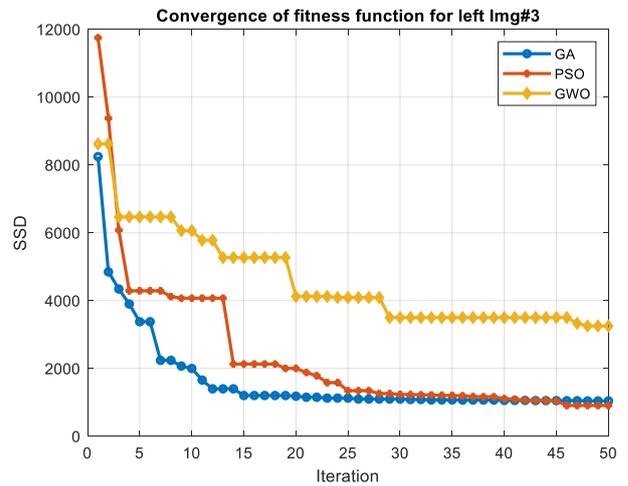


Fig. 9. Convergence of fitness function for left Img#3

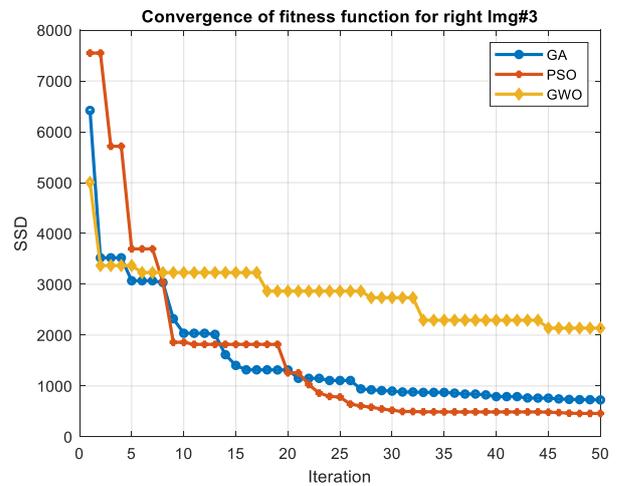


Fig. 10. Convergence of fitness function for right Img#3

For convergence speed comparison of BIO methods image 3 is selected due to the best geometrical alignment (Fig. 9 and Fig. 10). In Fig. 9 results suggest slightly faster convergence speed for GA with similar minimal FF value as PSO, while GWO is heavily outperformed. In Fig. 10 convergence speed is almost identical and the best FF value is in favor of PSO.

Reported results are procured in MATLAB software package running on workstation with Intel i7-7500U 2.7 GHz processor and 16 GB of RAM.

## VI. CONCLUSION

In this study, an experimental comparison of three Biologically Inspired Optimization (BIO) methods (Genetic Algorithm – GA, Particle Swarm Optimization – PSO, and Grey Wolf Optimizer - GWO) for Image Registration (IR) is made on a total of 24 images with different initial displacements. The comparison results demonstrate a successful application for minor initial displacements without a change in the orientation of the mobile robot RAICO. The comparison is made on images of manufacturing entities made in a laboratory model of a manufacturing environment. Considering experimental results, the fitness function does not significantly converge after a specific number of iterations, therefore additional stopping criteria can be defined in order to reduce the time required for the optimization process. With the implementation of different strategies for optimization parameters selection we expect to improve achieved results, which represents one of the future work directions. Further work could include an evaluation of other BIO methods for IR, and assessment of various fitness functions.

## ACKNOWLEDGMENT

This research was supported by the Science Fund of the Republic of Serbia, grant No. 6523109, AI - MISSION4.0, 2020-2022.

## REFERENCES

[1] B. Siciliano, K. Oussama, "Visual Servoing," in *Springer handbook of robotics*, Berlin, Germany: Springer, 2016, ch. 34, pp 841-866.

- [2] P. Corke, *Robotics, vision and control: fundamental algorithms in MATLAB*, 2nd ed., Berlin, Germany: Springer, 2017.
- [3] G. Silveira, E. Malis, "Visual servoing from robust direct color image registration," *IEEE/RSJ. International Conference on Intelligent Robots and Systems*, St. Louis, USA, pp. 5450-5455, Oct, 2009.
- [4] M. Irani, P. Anandan, "About direct methods," *International Workshop on Vision Algorithms*. Springer, Berlin, Heidelberg, Sept, 1999.
- [5] G. Silveira, E. Malis, "Direct visual servoing: Vision-based estimation and control using only nonmetric information," *IEEE Transactions on Robotics*, vol. 28, no.4, pp. 974-980, Apr, 2012.
- [6] B. Zitova, J. Flusser, "Image registration methods: a survey," *Image and vision computing*, vol. 21, no. 11, pp. 977-1000, Oct 2003.
- [7] L. Rundo, A. Tangherloni, C. Militello, M.C. Gilardi, G. Mauri, "Multimodal medical image registration using particle swarm optimization: A review," *IEEE. Symposium Series on Computational Intelligence (SSCI)*, Athens, Greece, pp. 1-8, Dec, 2016.
- [8] M.P. Wachowiak, R. Smolřková, Y. Zheng, J.M. Zurada, A.S. Elmaghaby, "An approach to multimodal biomedical image registration utilizing particle swarm optimization," *IEEE. Transactions on evolutionary computation*, vol. 8, no. 3, pp. 289-301, Dec, 2004.
- [9] H. Talbi, M. Batouche, "Hybrid particle swarm with differential evolution for multimodal image registration," *IEEE International Conference on Industrial Technology*, Hammamet, Tunisia, vol. 3, pp. 1567-1572, Dec, 2004.
- [10] F. L. Seixas, L. S. Ochi, A. Conci, D. M. Saade, "Image registration using genetic algorithms," *Proceedings of the 10th annual conference on Genetic and evolutionary computation*, Atlanta, USA, pp. 1145-1146, Jul, 2008.
- [11] A. Daly, H. Yazid, N. E. B. Amara, A. Zrig, "Multimodal image registration using multiresolution genetic optimization," *7th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT)*, Hammamet, Tunisia, pp. 557-562, Dec, 2016.
- [12] S. Mirjalili, S. M. Mirjalili, A. Lewis, "Grey Wolf Optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, Mar, 2014.
- [13] M. Petrović, A. Mystkowski, A. Jokić, L. Đokić, Z. Miljković, "Deep Learning-based Algorithm for Mobile Robot Control in Textureless Environment," *IEEE 15th International Conference Mechatronic Systems and Materials (MSM 2020)*, Bialystok, Poland, Jul, 2020.
- [14] M. Mitić, N. Vuković, M. Petrović, J. Petronijević, A. Diryag, Z. Miljković, "Bioinspired metaheuristic algorithms for global optimization," *Proceedings of the 5th International Conference on Information Society and Technology (ICIST 2015)*, Kopaonik, Serbia, pp. 38-42, Mar, 2015.
- [15] M. Petrović, Z. Miljković, B. Babić, "Optimization of operation sequencing in CAPP using hybrid genetic algorithm and simulated annealing approach," *Proceedings of the 11th International Scientific Conference MMA*, Novi Sad, Serbia, pp. 285-288, Sep, 2012.
- [16] M. Petrović, Z. Miljković, "Grey Wolf optimization algorithm for single mobile robot scheduling," *Proceedings of 4th international conference on electrical, electronics and computing engineering, IcETRAN*, Kladovo, Serbia, Jun, 2017.

# Experimental Testbed System for Building and Evaluation of Artificial Mind with Intelligent Robots ó Concept and Methodology

Aleksandar Rodi , Milo–Jovanovi , Dra–ko Furundfi

**Abstract** Intelligent beings possess a mind that allows them to be aware of the environment, experience, to think, learn, feel and conclude that they are aware of their emotions, understand others and control own feelings. An increasing interest in emotion can be seen in the behavioral, biological and social sciences but also in AI and intelligent systems. Research over the last two decades suggests that many phenomena, ranging from individual cognitive processing to social and collective behavior, cannot be understood without taking into account affective determinants such as motives, attitudes, moods, and emotions. The emerging field of affective science seeks to bring together the disciplines which study the biological, psychological and social dimensions of affect. The fact that emotions are considered to be essential to human reasoning suggests that they might play an important role in intelligent robots as well. The research objectives of the paper are focused on developing of an experimental testbed system for studying, modeling and practical implementation of AI algorithms with robots, which are related to scenario-driven and emotion-driven behavior (EDB). Aiming to this goal, the project requires a comprehensive data acquisition from humans, using an experimental AI-platform. Based on experimentally acquired data it will be accomplished a knowledge generalization and developed a knowledge-based model of human EDB using machine learning and fuzzy inference models. The developed models will be implemented in robot controller to enable facial gestures and speech reproduction. The project expects significant theoretical impacts on studying and modeling of human EDB, strongly modulated by personality traits. The project has a major impact on further development of AI algorithms with intelligent systems, for building smart personal devices characterized by a higher level of mind embodiment in IT industry and potential applications in medicine for therapy of children with developmental disorders.

**Index Terms**—Artificial mind, scenario-driven behavior, emotion-driven behavior, emotion state space.

## I. INTRODUCTION

Intelligent beings possess a mind<sup>1</sup> that allows them to be aware of the world around them, their experience, to think,

Aleksandar Rodi with Institute Mihajlo Pupin, Robotics department, Volgina 15, 11060 Belgrade, Serbia (e-mail: [aleksandar.rodic@pupin.rs](mailto:aleksandar.rodic@pupin.rs)).

Milo–Jovanovi with Faculty of computing, Union university, Knez Mihajlova 6, 11000 Belgrade, Serbia, (e-mail: [mjovanovic@raf.rs](mailto:mjovanovic@raf.rs)).

Dra–ko Furundfi with IMP Automatika, Volgina 15, 11060 Belgrade, Serbia (e-mail: [drasko.furundzic@pupin.rs](mailto:drasko.furundzic@pupin.rs)).

<sup>1</sup> According to the definition [1], the *mind* represents the element or complex of elements in an individual that feels, perceives, thinks, wills and reasons. Also, the mind means the organized conscious and unconscious adaptive mental activity of an organism.

learn, feel and conclude that they are aware of own emotions, control emotions and understand feelings of other people. An increasing interest in emotion can be seen in the behavioral, biological and social sciences but also more and more in AI and with intelligent systems (advanced robots). Research over the last two decades suggests that many phenomena, ranging from individual cognitive processing to social and collective behavior, cannot be understood without taking into account affective determinants (i.e. motives, attitudes, moods, and emotions) [2]. The emerging field of affective science<sup>2</sup> seeks to bring together the disciplines which study the biological, psychological and social dimensions of affect [3]. The fact that emotions are considered to be essential to human reasoning suggests that they might play an important role in intelligent systems such as autonomous robots.

To function in a complex and unpredictable physical and social environment humans are faced with applying their physical and intellectual resources to realize multiple goals in an intelligent and flexible manner. Two distinct and complementary information processing systems ó *cognition* and *emotion* enable humans achieving of these goals by operating simultaneously [4]. The cognitive system is responsible for interpreting and making sense of the world, whereas the emotion system is responsible for evaluating and judging events to assess their overall value with respect to the human (e.g. positive or negative, desirable or undesirable, hospitable or harmful, etc.) [4]. When operating in the proper balance, the emotion system modulates the operating parameters of the cognitive system and the body to improve the overall mental and physical performance of the human. The scientific literature documents the beneficial effect of emotion on creative problem solving, attention, perception, memory retrieval, decision-making, learning, etc. The fact is that too much emotion can hinder intelligent thought and behavior, however, too little emotion is even more problematic [5] being the people with a high cognitive capabilities and low emotional intelligence are rather inflexible in social settings and consequently are often useless for their environment (society, work or family).

<sup>2</sup> The *affective science* is scientific study of emotions or affects [2]. This includes the study of emotion expression, emotional experience and the recognition of emotions in others. In particular the nature of feeling, mood, emotionally driven behavior, decision making, attention and self-regulation, as well as the underlying physiology and neuroscience of the emotions.

Amidst the fourth industrial revolution, social robots are resolutely moving from fiction to reality. With sophisticated artificial agents becoming ever more ubiquitous in daily life, researchers across different fields are grappling with the questions concerning how humans perceive and interact with these agents and the extent to which the human brain incorporates intelligent machines into our social milieu. This paragraph surveys and discusses the latest findings, current challenges and future directions in neuroscience and psychology-inspired human-robot interaction (HRI) that are in focus of attention in this project. Critical questions that have to be explored from a transdisciplinary perspective when think about the artificial mind (eMIND) are: (i) technical solutions for HRI, (ii) development and learning for HRI, (iii) robots as a tool to study social cognition, and (iv) moral and ethical implications of HRI. Integrating findings from diverse but complementary research fields, including social and cognitive neurosciences, psychology, artificial intelligence, computer science and robotics, the contributions showcase ways in which research from disciplines spanning biological sciences, social sciences and technology deepen our understanding of the potential and limits of emotion-intelligent robotic agents in human social life.

Main scientific and technological (S/T) objectives of the project presented in the paper concerns with building attributes of Artificial Mind i.e. Electronic Mind (eMIND) especially focused to development new algorithms of emotional intelligence (EI) aimed to develop a robot EI-controller capable to manage and use emotions, manage relationship with others and increase the device autonomy.

## II. THEORETICAL BACKGROUND AND METHODOLOGY

The scientific fundaments that are used for development of knowledge-based model of human emotionally-driven i.e. scenario-driven behavior is the Meyers-Briggs theory of personality [8] widely used in the social sciences (e.g. psychology, sociology, etc.). According to this theory, the personality type and temperament of an individual dominantly determine human psychological (affective and social) behavior. The human personality and temperament are briefly explained in the text to follow.

### A. Personality Types

Personality psychology is a branch of psychology that studies personality and its individual differences [9]. According to the theory, "personality" is a dynamic and organized set of properties possessed by a person that uniquely influences individual cognitions, emotions, motivations, and behavior in various situations. Personality also refers to the pattern of thoughts, feelings, social adjustments, and behaviors consistently exhibited over time that strongly influences one's expectations, self-perceptions, values, and attitudes. It also predicts human reactions to other people, problems, and stress [10, 11]. This scientific discipline uses the Myers-Briggs Type Indicator (MBTI) assessment as a psychometric questionnaire designed to measure psychological preferences in how people perceive the world and make decisions [12]. The MBTI sorts psychological differences into four opposite pairs,

i.e. dichotomies (Fig. 1): Extravert-Introvert (E-I), Sensing-Intuitive (S-N), Feeling-Thinking (F-T) and Perceiving-Judging (P-J). That results in 16 possible psychological types (possible combinations). None of these types are better or worse. However, Briggs and Myers theorized that individuals naturally prefer one overall combination of type differences [12]. In the same way that writing with the left hand is hard work for a right-hander, so people tend to find using their opposite psychological preferences more difficult, even if they can become more proficient (and therefore behaviorally flexible) with practice and development. The 16 personality types [13] are typically referred to by an abbreviation of four letters, e.g. ESTJ, INTJ, ISTP, etc.

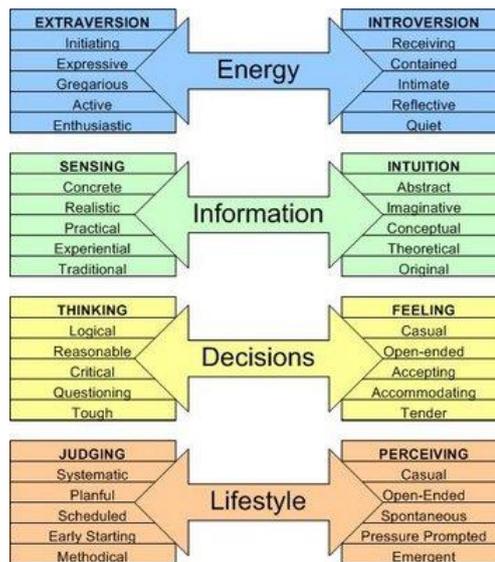


Fig. 1. Graphical presentation of the Meyers-Briggs theory of personality [8]

### B. Four Temperaments

Under the notion of temperament it is assumed in psychology, unlike to the term of personality, the individual kinds of the psyche traits that determine dynamics of human psychological activity [14]. The temperament traits are expressed in an even manner in any activity nevertheless to its content, goals and motives, remaining invariant in the later years and which, in their interconnections, characterize the type of temperament [15]. There are four temperaments (Fig. 2): (i) Sanguine (pleasure-seeking and sociable), (ii) Choleric (ambitious and leader-like), (iii) Melancholic (introverted and thoughtful), and (iv) Phlegmatic (relaxed and quiet).

The methodology we are to use in this project should allow a physical embodiment (materialization) of artificial (electronic) mind (eMIND) with intelligent systems, such as robots, based on application of the Meyers-Briggs personality theory and others taken from the social sciences of psychology and sociology. Generally, the human behavior is affected by various, generally not measurable, factors such as genetic factors (personality type, temperament and human character), external factors (family education, company/ friends, education, economic and social status, etc.) and interior stimuli (health condition, fatigue, stress level, current

emotional state, etc.). It is well known that people often react in a different manner to the same excitation event (trigger) because they differ in personality and character but also because of different influence from their social environment during a lifetime.

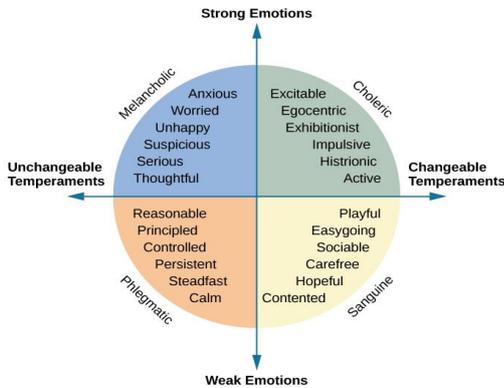


Fig. 2. Graphical presentation of the Four Temperaments Theory

The methodology to be applied in this project can be described briefly as follows. The model of human affective and social behavior will be derived based on learning scenario-driven i.e. emotion-driven behavior and personality traits from humans using experimental methods and on-line questionnaires (quiz). The obtained model of human psychological (cognitive) behavior will be implemented in a robot emotionally intelligence controller (EI-controller) to enhance better human-robot understanding and empathy. The personal data will be collected from a large number of human subjects (examinees), processed and used for extraction and generalization of knowledge about human emotional and social behavior. The raw data will be stored in the data-base DB-1 and processed data will be saved in the knowledge-base DB-2 (see Fig. 3).

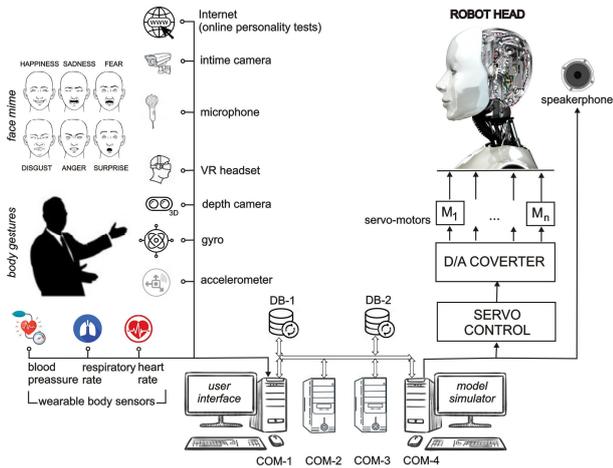


Fig. 3. The high-level system description presenting of corresponding functional modules. COM-1: Command interface and sensor-data acquisition. COM-2: Classifier and interpreter of human personality traits, trigger event types, interior and exterior stimuli. COM-3: Model of human emotionally-driven and scenario-driven behavior. COM-4: Robot EI-controller and simulator of human-inspired affective behavior. DB-1: Data-base i.e. raw data acquired from human examinees; DB-2: Data-base that contains processed row data obtained from classifiers.

Overall experimental testbed system is presented in Fig. 3. A specially designed robotic (animatronic) head (Fig. 4), with the image of Nikola Tesla inventor, is planned to be developed for the needs of this project. With this project, we want to achieve a certain symbolism and creative contribution in addition to the research goals.

Human-like affective reactions are based on three dominant excitation signals as the information carriers: (i) the trigger of behavior that arouses different psychological reactions; (ii) the profiler of behavior that shapes event-driven emotional state (ES) to fit personality profile of the individual; (iii) the behavior booster that gain affective response of individual. The high-level block diagram of the cognitive emotional model is shown in Fig. 4. Fuzzy blocks in the knowledge-based model shown in Fig. 4 will be designed in such a way to have an appropriate membership functions/parameters (type of function, focus of figure, inflection points, etc.), adjusted based on the data acquired from groups of subjects of human examinees.

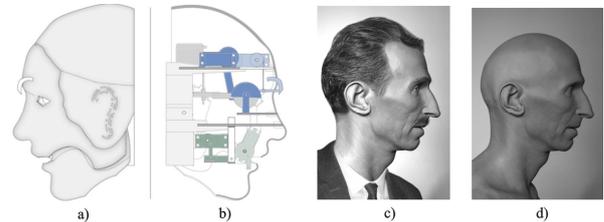


Fig. 2. Model of the mechatronic head to be developed in the project. a) 3D solid model of the mechatronic head; b) Presentation of the face mime mechanism; c) 3D model of the mask of Mr Nikola Tesla; d) The trial version of the hyper-realistic mask of Mr Nikola Tesla manufactured from the silicon rubber (Institute Mihajlo Pupin, 2019).

Several machine learning techniques (classification algorithms), fuzzy inference models and different optimization algorithms will be applied for synthesis of the human-inspired model of scenario-driven and emotion-driven behavior. The derived model of human psychological behavior will be implemented in the robot controller in the form of emotion intelligence EI-controller. For this purpose, the special hardware platform (Fig. 3) with appropriate software interface will be developed and used for research and testing new algorithms of AI, especially which regards to phenomena of affective and social behavior.

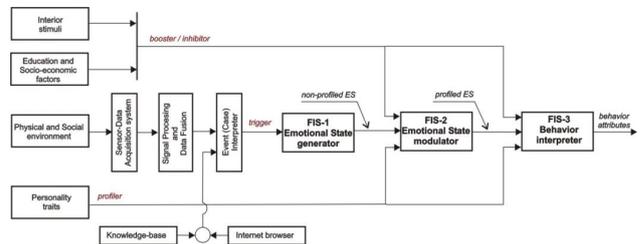


Fig. 4. Block-scheme of a three-stage scenario-driven and/or emotion-driven behavior model with fuzzy blocks: behavior generator, modulator and interpreter.

The personal data will be acquired from subjects in several ways: (i) by filling in existing on-line questionnaire (quiz) [13, 15] for determining a type of personality and temperament, (ii) by filling in for the purposes of this project specially prepared, thematic questionnaires that detect external (socioeconomic) as well as interior (which determine the current physical and mental state) factors which also influence cognitive behavior of the individual. Human examinees will be asked to give honest answers and to clearly quantify (to scale from 0-100%) the degree of influence of some causative factors on their behavior; (iii) by completing the appropriate questionnaire which can quantitate the degree of emotional experience of the excitation events descriptively imposed, (iv) by measuring the physical signals, using appropriate sensors and measuring wearable body devices ([16], Fig. 1), and (v) by presenting the excitation (trigger) events to human examinees in virtual reality.

### III. IMPLEMENTATION

The research planned in this paper does not focus only to the basic emotions. It uses emotion states such as valence and arousal making a dimensional space. In such an emotional space, personality traits are represented in the continuous domain. It is enabled by use of the proposed structure of the EI-model, conceptually set as presented in the block scheme in Fig. 4, with an emotion state modulator that makes fine resolution of the personality traits in the emotion space. Proposed model of human affective behaviour gives a wide degrees of freedom of representation robot's behaviour. Because of that it is suitable for implementation within the robot EI-controller to enable recognition of emotions of others as well as robot's self-tuning of its affective and social behavior.

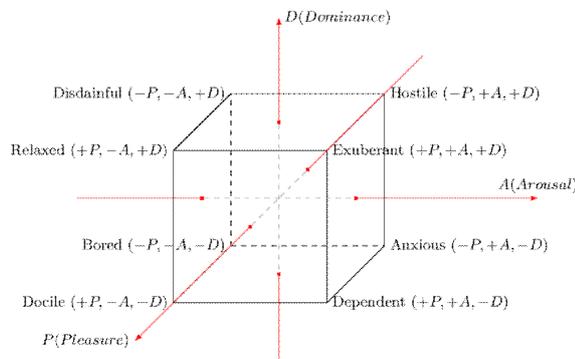


Fig. 5. Pleasure-Arousal-Dominance (PAD) model of human emotion state. Presentation of human personality traits in the 3-dimensional Emotion Space

As in the natural sciences, where there are dimensions such as mass, length, temperature, time, etc., researchers in the social sciences have developed similar dimensional spaces (emotional space of state). They serve to show, in a convenient way in a continuous space, different human emotional states. These dimensions are commonly referred to as emotional space. Points in this space (Fig. 5) determine individuals, segments or regions in this space define a certain

type of personality traits, while straight lines (coordinate axes) in this space define different dimensions of personality. There are several accepted ES models. In the research in this paper, the so-called a three-dimensional PAD space of emotional state, derived from the combination of the words Pleasure-Arousal-Dominance, is used. This space was defined by Russel and Mehrabian [17].

Briefly, the system shown in Fig. 3 uses various information (obtained from the sensory-visual-measuring system) from its physical and social environment to recognize the personality type of the interlocutor and to respond in an adequate way that implies an appropriate level of empathy and socially acceptable behavior. The system uses previously processed and priority prepared data, which are stored in the DB-2 database (Fig. 3). They refer to the characteristics of different personality types and their behavior in ordinary, everyday, life situations (events). These data were experimentally collected from a large number of subjects and after their processing and classification were placed in the database of the robotic controller. Using appropriate hardware resources and a rich database, the robotic controller applies the appropriate software application interface (EI-model that represents a novelty of this research), shown in the block diagram of Fig. 4. The functional block FIS-2 determines the dimensions of the personality in the state space shown in Fig. 5. As a final result, the robotic controller generates appropriate behavior (verbal and nonverbal manifestations) in response to a trigger event or situation.

### IV. CONCLUSION

Despite of multitude of valuable research works, human brain as well human emotion-driven and scenario-driven (affective and social) behavior is still not enough explored and requires new more generic solutions. Majority of the existing solutions are based on internal learning approaches aim to building of appropriate developmental mechanisms and software architectures intended for acquiring new skills and new knowledge in embodied machines. These models develop systems that can recognize, interpret, process, and simulate human emotions (affects) based on perceived event characteristics. Our EI-model considers problem of human behavior in a more comprehensive way than other known models by taking into account greater number of influencing factors (arguments) which determine the affective and social behavior of the individuals. The solution is based on use of a broad knowledge base comprising information about the personality profile of the subjects being interviewed, external factors affecting development of the emotional individuals (e.g. socio-economic factors, education, family education, etc.), internal stimuli that enhance or inhibit emotional reactions (e.g. health condition, stress, feeling of satisfaction, sense of personal achievement, etc.) and the character of the events that trigger affective reactions and moods. The problem is going to be solved by application of the machine learning algorithms and fuzzy inference models. In the project it will be synthesized a unique generic knowledge-based model of human affective and social behavior. The model represents deterministic, non-linear, multi-input and multi-output

(MIMO) system of arbitrary complexity. Model will be designed as a unique hybrid neuro-fuzzy system that get together best characteristics of both methods of efficient learning structures and fuzzy inference systems. The proposed generic model can be upgraded or additionally extended by adding new arguments in the existing model structure and by extending number of fuzzy rules in the model. The proposed fuzzy model will be modular and consists of three blocks (Fig. 4): behavior generator, behavior modulator and moods interpreter. The mentioned fuzzy blocks will be experimentally tuned by means of the knowledge-base developed in the project.

#### A. Expected novelties

To summarize the following novelties in the project are expected to be delivered: (i) original hardware platform with software interface for extensive data-acquisition, sensor-data fusion and signal processing of human affective and social behavior, (ii) experimental testbed developed for R&D and verification of new AI algorithms, (iii) multi-stage knowledge-based model of human-inspired scenario-driven and emotion-driven behavior, and (vi) robot EI-controller that enables embodiment of robot artificial mind (eMIND). This feature enhances cognitive human-robot interface cHRI characteristic for smart personal devices.

#### REFERENCES

- [1] Mind meaning. <https://www.merriam-webster.com/dictionary/mind>. Last retrieved on July, 30<sup>th</sup>, 2020
- [2] Affective science. [http://en.wikipedia.org/wiki/Affective\\_science](http://en.wikipedia.org/wiki/Affective_science). Last retrieved on July, 30<sup>th</sup>, 2020
- [3] Francis, A.G., Mehta, M.J., Ram, A.: *Handbook of Research on Synthetic Emotion and Sociable Robotics: New Applications in Affective Computing and Artificial Intelligence*. Ed. J. Vallverdú, D. Casacuberta. Chapter XX: Emotional Memory and Adaptive Personalities. Information science reference, Hershey-New York, pp. 391-412 (2009).
- [4] Breazeal, C. "Function Meets Style: Insights from Emotion Theory Applied to HRI", *IEEE Transactions in Systems, Man, and Cybernetics, Part C*, 34(2), 187-194, 2003
- [5] Damasio, A., *Descartes Error: Emotion, Reason, and the Human Brain*. G.P Putnam's sons, New York, 1994
- [6] Testbed meaning. <https://www.thefreedictionary.com/Testbed>. Last retrieved on July, 30<sup>th</sup>, 2020
- [7] Emotional intelligence. [http://en.wikipedia.org/wiki/Emotional\\_intelligence](http://en.wikipedia.org/wiki/Emotional_intelligence). Last retrieved on July, 30<sup>th</sup>, 2020
- [8] Mayer, J.D. and Salovey, P. "What is emotional intelligence?" In: P. Salovey & D. Sluyter (Eds.), *Emotional development and emotional intelligence: Implications for educators*, pp. 3-31, New York: Basic Books, 1997
- [9] Petrides, K.V., Pita, R., Kokkinaki, F. "The location of trait emotional intelligence in personality factor space", *British Journal of Psychology*, 98, 273-289, 2007
- [10] Myers-Briggs, I. and Myers, P. *Gifts Differing: Understanding Personality Type*. Mountain View, CA: Davies-Black Publishing. 1995
- [11] Personality psychology: [http://en.wikipedia.org/wiki/Personality\\_psychology](http://en.wikipedia.org/wiki/Personality_psychology). Last retrieved on July, 30<sup>th</sup>, 2020
- [12] Myers, D. G. *Psychology* (9th ed.). New York: Worth Publishers. 2010
- [13] Winnie, J.F. and Gittinger, J.W. "An introduction to the personality assessment system", *Journal of Clinical Psychology, Monograph Supplement*, 38, pp. 1-68, 1973
- [14] MBTI: <http://en.wikipedia.org/wiki/Myers-Briggs-Type-Indicator>. Last retrieved on July, 30<sup>th</sup>, 2020
- [15] On-line personality test: <http://www.16personalities.com/free-personality-test>. Last retrieved on July, 30<sup>th</sup>, 2020
- [16] Temperament types: [http://en.wikipedia.org/wiki/Four\\_temperaments](http://en.wikipedia.org/wiki/Four_temperaments). Last retrieved on July, 30<sup>th</sup>, 2020
- [17] On-line 4 temperament test: <https://temperamenttest.org/en-gb/>. Last retrieved on July, 30<sup>th</sup>, 2020
- [18] Empatica E4 wristband sensor. <https://www.empatica.com/en-int/research/e4/>. Last retrieved on July, 30<sup>th</sup>, 2020
- [19] J. A. Russel, A. Mehrabian, "Evidence for a three-factor theory of emotions", *Journal of research in Personality*, Vol. 11, No. 3, pp. 273-294, 1977

# Cyber-attack detection method based on RNN

Dušan Nedeljković, Živana Jakovljević, *Member, IEEE*

**Abstract**—Current and forthcoming market requirements bring huge challenges to today manufacturing. Answer to the changing demands and high product variety is found in the integration of the Internet of Things (IoT) and Cyber-Physical Systems (CPS) into industrial plants. CPS as smart devices capable of data processing and information exchange enable fast adaptation of manufacturing resources to production of diversified products. Nevertheless, fully implemented internet communication at factory shop floor opens up a whole new area for potential cyber-attacks. The consequences of attacks can have a negative influence on the system or even endanger human lives. Therefore, defence techniques must be developed to ensure a high level of protection. Early detection of cyber-attacks is crucial to minimize or completely avoid the negative effects of the attack and keep the system safe and reliable. In this work, we propose an attack detection method based on deep learning approach. We explore the application of several deep learning architectures based on Simple Recurrent Neural Networks (Simple RNN) and Long Short-Term Memory (LSTM) based RNN for generation of the detection mechanisms tailored to the concrete process. Our method was experimentally verified using real world data and it proved to be effective, as it detected all considered attacks without false positives.

**Index Terms**—Cyber security; Cyber Physical Systems; Internet of Things; Deep learning; Recurrent Neural Network.

## I. INTRODUCTION

Traditional industrial systems are experiencing changes driven by market requirements, especially in terms of manufacturing systems adaptability to changing demands that should be achieved along with the preservation of their efficiency and productivity. Such changes are seeking smart components (sensors, actuators...) in the form of physical devices with integrated computation and communication capabilities - Cyber-Physical Systems (CPS) [1, 2]. Within an industrial plant, smart devices interact with a central control system and with each other, share information over the Internet (usually wireless), creating and exchanging large amounts of data through Industrial Internet of Things (IIoT) [3, 4]. Data generated through fully implemented communication are stored in clouds or on the edge and used to make business related conclusions and timely control decisions. The integration of IoT and CPS into plant represents the core of the concept called Industry 4.0 [5]. This approach was created to meet performances, availability,

flexibility requirements, as well as a new level of product individualization. However, openness and adopting the internet as the main way of communication in traditionally isolated industrial control systems introduce new risks and make these systems exposed to various types of cyber-attacks. Consequences of the attacks can degrade system performance, cause serious damages and production losses, or even affect human lives [6]. Therefore, the development of cyber-attack detection techniques represents one of the main industrial security issues today. To reduce or completely avoid the negative effects of attacks, they must be timely detected despite the fact they are trying to stay stealthy.

In general, cyber-attacks can be roughly divided into two categories: Denial of Service (DoS) attacks and deception attacks [7]. DoS attacks interfere with the data flows, making them temporarily or permanently unavailable [8]. DoS attacks are not necessarily system-dependent, i.e. do not require prior knowledge about the system they attack. Deception attacks, on the other hand, affect data integrity by injecting malicious data with an aim to completely change system behaviour [9]. Unlike DoS, deception attacks need knowledge about system resources, in order to attack the most sensitive system parts, while remaining stealthy. Mathematically, DoS attacks can be described with  $\bar{x}_i \in \emptyset$ , and deception attacks with  $\bar{x}_i = x_i + x_i^d$ , where  $\bar{x}_i$ ,  $x_i$  and  $x_i^d$  represent received, measured and data injected by attackers, respectively [10].

The appearance of new attacks also triggers the development of new defence techniques. Today, there are a number of techniques that can be split into two categories: data-centric and design-centric [11]. Data-centric techniques are based on the collected data, while design-centric are oriented to system analytical models and its control algorithms. For example, the detection of false injected data [12] and compromised sensory data [13] were performed using a design-centric method based on cumulative sum (CUSUM). Also, a number of data-centric detection techniques were developed, such as 1D convolutional neural networks [14], and autoregression modelling and control limits [15] that were deployed for anomaly detection in Industrial Control Systems.  $\epsilon$ -insensitive support vector regression ( $\epsilon$ -SVR) was utilized for this security issue in the system with distributed control [16].

In this work, we propose a data-centric method for cyber-attacks detection in continuous time controlled systems. The method is based on deep learning model of normal system behaviour, created using Recurrent Neural Networks (RNN). The performance of the method is evaluated on a signal from the real-world system with several cyber-attacks. In particular, we have considered data obtained from a smart actuator as a part of the system with distributed control. The idea of

Dušan Nedeljković is with the Faculty of Mechanical Engineering, University of Belgrade, Kraljice Marije 16, 11020 Belgrade, Serbia (e-mail: dnedeljkovic@mas.bg.ac.rs).

Živana Jakovljević is with the Faculty of Mechanical Engineering, University of Belgrade, Kraljice Marije 16, 11020 Belgrade, Serbia (e-mail: zjakovljevic@mas.bg.ac.rs).

utilizing RNN for this purpose is based on their capability to use sequences and memorize the information obtained in the previous calculations. The memory from prior computation allows RNN to better understand data dependencies and to create a good prediction. Additionally, we compare the results obtained using different RNN architectures and select the best model according to the defined criteria.

The remainder of the paper is structured as follows. Section 2 briefly outlines used RNN architectures, whereas Section 3 refers to the developed method for signal attacks detection. In Section 4 we represent the results of the proposed method evaluation using data from the real-world application with inserted attacks. Conclusions and future work guidelines are provided in Section 5.

## II. RECURRENT NEURAL NETWORK

Recurrent Neural Networks differ from standard feed-forward neural networks by connection where the current output vector  $\mathbf{y}(t)$  depends not only on the present input  $\mathbf{x}(t)$ , but also on the recurrent input representing previous hidden state  $\mathbf{h}(t-1)$ . Thus, using history, network learns and understands the sequential nature of the data. Fed with the input and the previous hidden state vector, RNN cyclically computes output for every element of a sequence. Two types of RNNs are used in this paper: 1) Simple RNN, and 2) Long Short-Term Memory (LSTM).

### A. Simple RNN

Simple RNN also called Elman network [17], represents a fully-connected network with a feedback. The loop keeps the hidden state vector at a previous time step  $\mathbf{h}(t-1)$  and feeds it with the new input vector  $\mathbf{x}(t)$ , as shown in Fig. 1. Therefore, Simple RNN has the most general topology and most similar to the regular neural networks architectures.

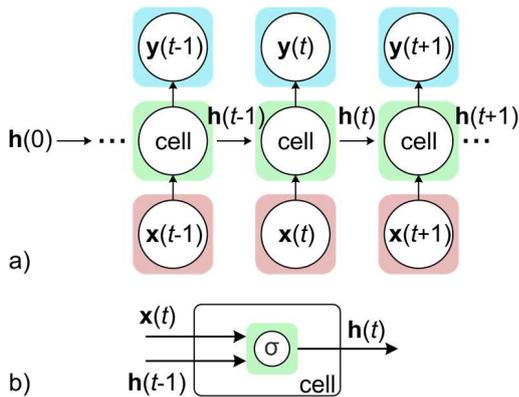


Fig. 1. Simple RNN: a) network architecture; b) cell(t) architecture

The initial value (usually set to 0) of the hidden state vector is denoted by  $\mathbf{h}(0)$ . Hidden state vector  $\mathbf{h}(t)$  at a time step  $t$  is calculated as follows:

$$\mathbf{h}(t) = \sigma(\mathbf{W}_{xh}\mathbf{x}(t) + \mathbf{W}_{hh}\mathbf{h}(t-1) + \mathbf{b}_h), \quad (1)$$

where  $\mathbf{b}_h$  is a bias vector,  $\sigma$  represents the activation function,  $\mathbf{W}_{xh}$  and  $\mathbf{W}_{hh}$  denote the input and hidden weight matrices, respectively. RNN output  $\mathbf{y}(t)$  is defined by the following equation:

$$\mathbf{y}(t) = \sigma(\mathbf{W}_{ho}\mathbf{h}(t) + \mathbf{b}_o), \quad (2)$$

where  $\sigma$  is output activation function,  $\mathbf{W}_{ho}$  represents output weight matrix, and  $\mathbf{b}_o$  is a bias vector.

### B. LSTM

LSTM is developed to overcome the vanishing gradient problem that in a number of cases makes regular RNN hard to train [18]. Recurrently connected modules called memory blocks present the basis of LSTM architecture. Each memory block consists of one or more memory cells connected in a certain way with a set of multiplicative units (gates), as shown in Fig. 2.

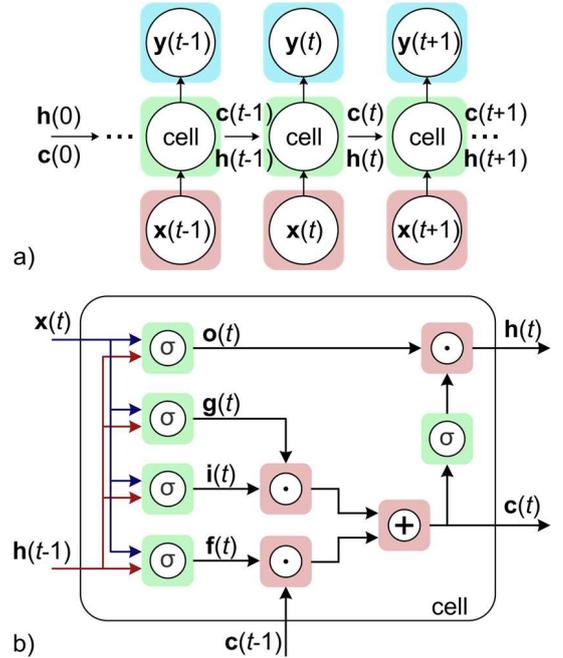


Fig. 2. LSTM: a) network architecture; b) cell (t) architecture

$\mathbf{h}(0)$  and  $\mathbf{c}(0)$  are usually set to 0 and represent the initial hidden state and cell state vectors, respectively. Input ( $i$ ), Output ( $o$ ) and Forget ( $f$ ) gates regulate when new information enters, select useful information as output and forget (remove) unnecessary information from the current cell state, respectively. All gates have a common task to prevent memory from perturbation by irrelevant inputs and outputs and thus ensure long term memory storage. LSTM block also contains candidate hidden state ( $g$ ), which is based on current input and the previous hidden state. Gates and hidden state vectors are computed in the following way:

$$\begin{aligned} \mathbf{i}(t) &= \sigma(\mathbf{W}_{xi}\mathbf{x}(t) + \mathbf{W}_{hi}\mathbf{h}(t-1) + \mathbf{b}_i), \\ \mathbf{f}(t) &= \sigma(\mathbf{W}_{xf}\mathbf{x}(t) + \mathbf{W}_{hf}\mathbf{h}(t-1) + \mathbf{b}_f), \\ \mathbf{o}(t) &= \sigma(\mathbf{W}_{xo}\mathbf{x}(t) + \mathbf{W}_{ho}\mathbf{h}(t-1) + \mathbf{b}_o), \end{aligned} \quad (3)$$

$$\mathbf{g}(t) = \sigma(\mathbf{W}_{xg}\mathbf{x}(t) + \mathbf{W}_{hg}\mathbf{h}(t-1) + \mathbf{b}_g).$$

Weight matrices  $\mathbf{W}_x$  and  $\mathbf{W}_h$  are divided into 4 parts, where each part belongs to corresponding gate:  $i$ ,  $f$ ,  $o$  and  $g$ . Other variables labelling is in accordance with Section II-A. The cell state is determined by its previous value  $\mathbf{c}(t-1)$ , candidate hidden state, input and forget gates.

$$\mathbf{c}(t) = \mathbf{f}(t) \odot \mathbf{c}(t-1) + \mathbf{i}(t) \odot \mathbf{g}(t), \quad (4)$$

where  $\odot$  represents element-wise multiplication of the vectors. Hidden state vector represents the output of LSTM cell:

$$\mathbf{h}(t) = \mathbf{o}(t) \odot \sigma(\mathbf{c}(t)). \quad (5)$$

LSTM output  $\mathbf{y}(t)$  is defined in the following way:

$$\mathbf{y}(t) = \sigma(\mathbf{W}_{hy}\mathbf{h}(t) + \mathbf{b}_y), \quad (6)$$

where  $\mathbf{W}_{hy}$  represents output weight matrix, and  $\mathbf{b}_y$  is a bias vector.

### III. SENSOR SIGNAL ATTACK DETECTION METHOD

The method that we propose in this paper consists of two phases: offline RNN training and online attack detection. The training process is based on sensory data recorded under normal conditions (without anomalies/attacks). The model generated during training describes proper sensor operating and represents the basis for signal prediction in attack detection part. The second step considers the difference between signal values estimated by model and measured sensor signal values (Fig. 3).

In our method, the current value of the sensor signal  $x(t)$  is predicted based on the sequence of previous  $z$  values  $x(t-z), \dots, x(t-1)$ . Thus, through the RNN training process ordered pairs are created:

$$\begin{aligned} (\mathbf{x}(t), \mathbf{y}(t)) \in \{ & ([x(1), \dots, x(z)], x(z+1)), \\ & ([x(2), \dots, x(z+1)], x(z+2)), \dots, \\ & ([x(n-z), \dots, x(n-1)], x(n)) \}, \end{aligned} \quad (7)$$

where  $\mathbf{x}(t)$ ,  $t \in [z+1, n]$  represents vector of input variables, and  $\mathbf{y}(t)$  denotes corresponding response. During training, a few parameters should be tuned to get the best possible RNN model. The sequence length  $z$  specifies the number of samples which determine the next predicted value. Batch size ( $bs$ ) defines the number of samples to work with before updating the weights. The network architecture is determined by the number of layers ( $nl$ ) and units per layer ( $ul$ ). A unique RNN model is obtained by varying one or more parameter values. The first criterion for selecting the best model is the minimum value of  $p$ , defined as follows:

$$p = \frac{1}{n-z} \sum_{t=z+1}^n |x(t) - \hat{x}(t)|. \quad (8)$$

Parameter  $p$  represents the mean absolute difference between real ( $x(t)$ ) and predicted ( $\hat{x}(t) = y(t)$ ) values of sensor readings over the test dataset. To make the model as simple as possible and to reduce training and testing time, the number of model parameters should not be too large. Furthermore, to keep the latency that is introduced during online application of detection mechanisms at acceptable level, the number of RNN model parameters is especially significant and should be as low as possible. Thus, a small number of layers in the network and a small number of units per layer represent the second criterion for model selection. However, it should be emphasized that insufficient number of layers and units may lead to an inaccurate model and erroneous attack detection thereof.

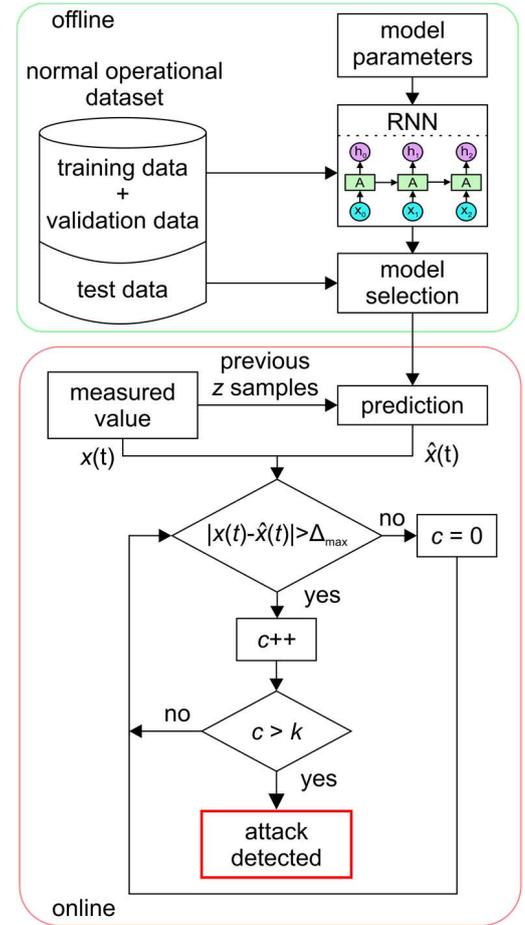


Fig. 3. Algorithm for sensor signal attacks detection

The selected RNN model, according to the defined criteria, represents the output from the offline phase, i.e. the input to the online phase. Online attack detection is based on the absolute difference between measured and predicted values. If error exceeds the detection threshold  $\Delta_{max}$  consecutively for  $k$  samples, the attack is present. The threshold value  $\Delta_{max}$  is defined as a sum of mean value  $\mu$  and standard deviation  $\sigma$  of discrepancy between measured and estimated values over the training data:

$$\Delta_{max} = \mu + \sigma, \quad (9)$$

where  $\mu$  and  $\sigma$  are:

$$\begin{aligned}\mu &= \frac{1}{n-z} \sum_{t=z+1}^n (x(t) - \hat{x}(t)), \\ \sigma &= \sqrt{\frac{1}{n-z} \sum_{t=z+1}^n (x(t) - \mu)^2}.\end{aligned}\quad (10)$$

#### IV. EXPERIMENTAL RESULTS

The proposed method for attacks detection is experimentally evaluated on electro-pneumatic positioning system [16] with control distributed on smart sensor and smart actuator. Smart actuator consists of linear rodless pneumatic cylinder supplied by air through electro-pneumatic air pressure regulator (output pressure in the range 2-6 bar) on one, and mechanically controlled air pressure regulator (constant output pressure of 4 bar) on the other side. As a part of the smart actuator, local controller – wireless node controls output pressure on the electro-pneumatic regulator.

On the other hand, the smart sensor represents linear encoder (placed along the cylinder) that is equipped with its own local controller – wireless node (LC2). LC2 obtains pulses from the encoder and determines the position of the piston. Corresponding to the desired position, LC2 generates a control signal (in the range 0-1) and transmits it to LC1 using wireless communication. Furthermore, LC1 converts the received signal value to the analog voltage in the range 0-10 V and sends it to electro-pneumatic regulator. Electro-pneumatic regulator gives at its output the pressure proportional to the input voltage. Air pressure difference between the two sides of the piston causes piston movement.

Sensor signals were recorded during normal system functioning, without attacks. In particular, the voltage between LC1 and electro-pneumatic air pressure regulator was acquired using characteristic piston trajectory that contained positions of 50, 400, 250, 400, and 100 mm. The defined trajectory was cyclically repeated 100 times. The data acquisition was performed with a sampling rate of 100 Hz, which led to a total of 400,000 records.

To find the optimal model of sensory data we used different architectures based on two selected RNN types. We have opted to use two RNN layers since a model with more than 2 RNN layers contains a large number of model parameters, whereas, a single layer cannot meet the required model performance. The architecture that has shown the best results (Fig. 4) for all models is: RNN layer (units) - Dropout (rate) - RNN layer (units) - Dropout (rate) - Dense (1).

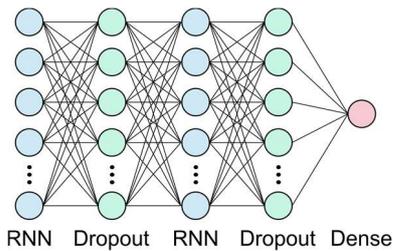


Fig. 4. The chosen network architecture

Dropout layer helps prevent overfitting by temporarily removing minimal units from the network with a rate from 0 to 1 (0 to 100% of all units). A Dense layer is a fully connected layer, where each input node is connected to each output node. Five RNN architecture related parameters were varied during the process of finding the optimal model. The considered values of RNN architecture parameters are given in Table I. In addition, 4 different batch sizes that affected the values of particular RNN parameters were explored.

This variation of parameters resulted in a total of 432 different models (216 models for both RNN types: Simple RNN and LSTM). During models training, the whole dataset is divided into training, validation and test part, with a ratio of 80/10/10 %, respectively. The validation and training losses were almost unchanged after 5 epochs, so this was selected as the optimal number of epochs during models training. Through the process of finding the most appropriate model, we used Adam optimizer with learning rate of 0.001 and rectified linear unit (ReLU) activation function. The cost function that we utilized was mean squared error (MSE).

TABLE I  
VARIED RNN PARAMETERS

Parameter	Value
sequence length ( $z$ )	2, 5, 10
dropout rate ( $dp$ )	0.05, 0.1
number of units in layer1 ( $ul1$ )	8, 16, 32
number of units in layer2 ( $ul2$ )	8, 16, 32
batch size ( $bs$ )	8, 16, 32, 64

The models were trained in Python using a Visual Code Studio with Keras and TensorFlow at the background. To test our proposed signal attack detection method and to choose an appropriate model, a number of attacks have been created. In this paper, we present three attacks of different types and duration. Attack 1 ( $A_1$ ) utilizes sinus function to generate  $x$  value, attack 2 ( $A_2$ ) increases  $x$  value linearly, whereas in attack 3 ( $A_3$ ) value of  $x$  is immediately set to 0. The following equations respectively define these three attacks:

$$A_1: x(t) = 0.5 + \sin(0.005 \cdot t), t = 1, 2, \dots, 1300$$

$$A_2: x(t) = x(t) + 0.00007 \cdot t + 0.0005 \cdot \text{rand}(), t = 1, 2, \dots, 400 \quad (11)$$

$$A_3: x(t) = 0, t = 1, 2, \dots, 500.$$

According to the first criterion (minimum value of  $p$ ), the best five models of both RNN types were chosen. The performances of the selected models are given in Table II where the best-performing models for both of the considered RNN types are highlighted; Fig. 5 presents histogram of absolute errors between measured and predicted values for these models.

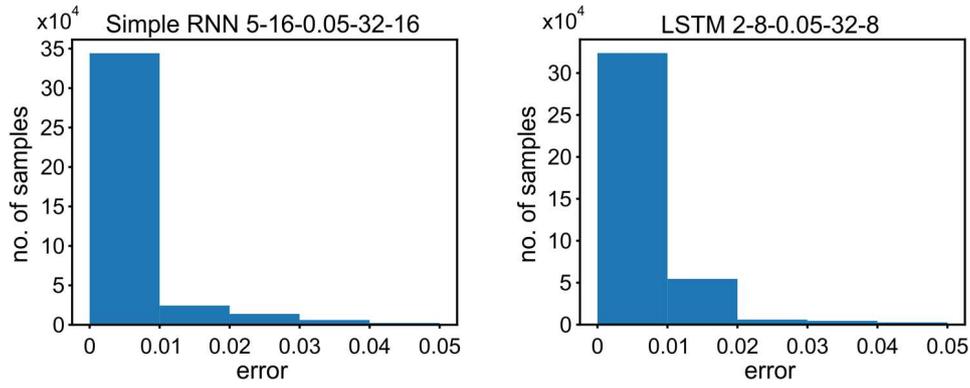


Fig. 5. Histograms of absolute errors between measured and predicted values for selected models; for the clarity of presentation, histograms do not contain 8084 (2.02%) samples whose errors are in the range [0.05, 0.98795] for Simple RNN and 7482 (1.88%) samples whose errors are in the range [0.05, 0.97650] for LSTM model.

TABLE II  
PERFORMANCES OF THE SELECTED RNN MODELS

RNN type- $z$ - $bs$ - $dp$ - $ul1$ - $ul2$	$p$	no. of param.	attacks detected	false positives	false negatives
<b>SimpleRNN-5-16-0.05-32-16</b>	<b>0.00650</b>	<b>1889</b>	<b>2/3</b>	<b>1</b>	<b>1</b>
SimpleRNN-5-64-0.05-16-16	0.00697	833	3/3	>10	0
SimpleRNN-10-32-0.05-32-16	0.00720	1889	3/3	>10	0
SimpleRNN-10-16-0.1-32-16	0.00813	1889	3/3	>10	0
SimpleRNN-5-64-0.05-32-16	0.00857	1889	3/3	>10	0
LSTM-5-8-0.1-32-16	0.00391	7505	1/3	0	2
LSTM-10-64-0.05-32-16	0.00457	7505	2/3	0	1
LSTM-5-8-0.1-32-32	0.00486	12705	3/3	6	0
LSTM-5-64-0.1-16-16	0.00487	3281	2/3	0	1
<b>LSTM-2-8-0.05-32-8</b>	<b>0.00500</b>	<b>5673</b>	<b>3/3</b>	<b>0</b>	<b>0</b>

From Table II it can be observed that LSTM-2-8-0.05-32-8 model can detect all three attacks without false-positive results. However, no model with Simple RNN architecture meets such requirement. Some of the models are not suitable for attack detection, although they provide an excellent prediction. The reason is that these models do not make difference between the attacks and signal under normal operating conditions. Simple RNN-5-16-0.05-32-16 is the model that is the closest to detecting all attacks without false positives when Simple RNN architectures are considered. Measured signal, predicted values, and detected attacks for two selected models are represented in Fig. 6. The input data and their prediction are shown in blue and green lines, respectively. Moments when the attack was detected, are marked with red \*. The detection method based on both models successfully detected attacks 1 and 3.

However, the method based on the Simple RNN model was not able to detect attack 2. Besides, this method also has one

false-positive result that is not shown in Fig. 6.

It can be observed from Fig. 6 that the LSTM-based method detected attack 3 more effectively than Simple RNN method, i.e. it required the smaller number of samples from the beginning of the attack to its detection resulting in lower attack detection latency. In the case of attack 1 detection, both methods proved to be equally effective. Therefore, model LSTM-2-8-0.05-32-8 is considered as the best.

## V. CONCLUSION

We have proposed a method for signal attack detection based on a prediction of signal value using deep learning algorithms. More precisely, we used two popular recurrent neural network architectures: Simple RNN and LSTM.

Deep learning models have been trained on a dataset obtained from an electro-pneumatic positioning system under normal conditions (without attacks). We have generated over 400 models by varying several parameters and according to

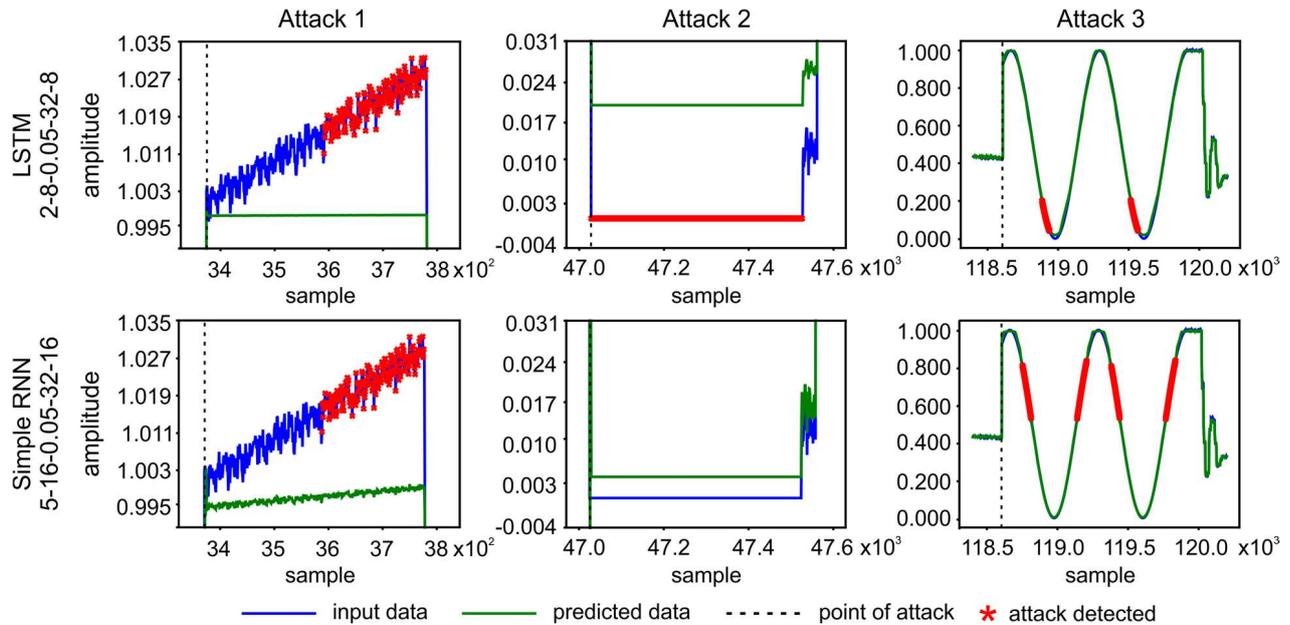


Fig. 6. Detected attacks

the defined criteria, we have selected the best models for each RNN architecture. For the method evaluation, we have created three different attacks. The method proved effective in cases of LSTM architecture as it detected all attacks without false positives.

Further research will focus on the implementation of the method in real-world on low level controllers of CPS, primarily on the electro-pneumatic positioning system. Additionally, the method will be tested on publicly available datasets with a number of different attacks.

#### ACKNOWLEDGMENT

This research was supported by the Science Fund of the Republic of Serbia, grant No. 6523109, AI - MISSION4.0, 2020-2022.

#### REFERENCES

- [1] L. Monostori, B. Kádár, T. Bauernhansl, S. Kondoh, S. Kumara, G. Reinhart, O. Sauer, G. Schuh, W. Sihn, K. Ueda, "Cyber-physical systems in manufacturing," *CIRP Annals*, vol. 65, no. 2, pp. 621-641, 2016.
- [2] Z. Jakovljevic, V. Majstorovic, S. Stojadinovic, S. Zivkovic, N. Gligorijevic, M. Pajic, "Cyber-physical manufacturing systems (CPMS)," Proc. 5th Int. Conf. Adv. Manuf. Eng. Technol. (NEWTECH 2017), Belgrade, Serbia, pp. 199-214, June 2017.
- [3] L. Atzori, A. Iera, G. Morabito, "The Internet of Things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787-2805, Oct. 2010.
- [4] Z. Jakovljevic, V. Lesi, S. Mitrovic, M. Pajic, "Distributing Sequential Control for Manufacturing Automation Systems," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 4, pp. 1586-1594, 2020.
- [5] H. Kagermann, W. Wahlster, J. Helbig, *Recommendations for implementing the strategic initiative INDUSTRIE 4.0*, 2013. [Online]. Available: <http://www.acatech.de>
- [6] Z. Jakovljevic, V. Lesi, M. Pajic, "Attacks on Distributed Sequential Control in Manufacturing Automation," *IEEE Transactions on Industrial Informatics*, 2020, doi: 10.1109/TII.2020.2987629
- [7] A. Teixeira, D. Pérez, H. Sandberg, K. H. Johansson, "Attack Models and Scenarios for Networked Control Systems," Proceedings of the 1st International Conference on High Confidence Networked Systems, Beijing, China, pp. 55-64, Apr. 2012.

- [8] S. Amin, A. A. Cárdenas, S. S. Sastry, "Safe and secure networked control systems under Denial-of-Service attacks," Proceedings of the 12th International Conference on Hybrid Systems: Computation and Control, San Francisco, USA, pp. 31-45, Apr. 2009.
- [9] Q. Zhang, K. Liu, Y. Xia, A. Ma, "Optimal Stealthy Deception Attack Against Cyber-Physical Systems," *IEEE Transactions on Cybernetics*, May 2019.
- [10] D. Ding, Q.-L. Han, Y. Xiang, X. Ge, X.-M. Zhang, "A survey on security control and attack detection for industrial cyber-physical systems," *Neurocomputing*, vol. 275, pp. 1674-1683, 2018.
- [11] H. S. Sánchez, D. Rotondo, T. Escobet, V. Puig, J. Quevedo, "Bibliographical review on cyber attacks from a control oriented perspective," *Annual Reviews in Control*, vol. 48, pp. 103-128, Sep. 2019.
- [12] Y. Huang, J. Tang, Y. Cheng, H. Li, K. A. Campbell, Z. Han, "Real-Time Detection of False Data Injection in Smart Grid Networks: An Adaptive CUSUM Method and Analysis," *IEEE Systems Journal*, vol. 10, no. 2, pp. 532-543, June 2016.
- [13] C. Murguía, J. Ruths, "CUSUM and chi-squared attack detection of compromised sensors," 2016 IEEE Conference on Control Applications (CCA), Buenos Aires, Argentina, pp. 474-480, Sept. 2016.
- [14] M. Kravchik, A. Shabtai, "Detecting Cyber Attacks in Industrial Control Systems Using Convolutional Neural Networks," Proceedings of CPS-SPC 18 Conference, Toronto, Canada, pp. 72-83, Oct. 2018.
- [15] D. Hadziomanović, R. Sommer, E. Zambon, P. H. Hartel, "Through the eye of the PLC: semantic security monitoring for industrial processes," Proceedings of 30th Annual Computer Security Applications Conference, New Orleans, USA, pp. 126-135, Dec. 2014.
- [16] D. M. Nedeljkovic, Z. B. Jakovljevic, Z. Dj. Miljkovic, M. Pajic, "Detection of cyber-attacks in electro-pneumatic positioning system with distributed control," 27th Telecommunications Forum (TELFOR), Belgrade, Serbia, art. no. 8971062, Nov. 2019.
- [17] J. Elman, "Finding Structure in Time," *Cognitive Science*, vol. 14, no. 2, pp. 179-211, Mar. 1990.
- [18] S. Hochreiter, J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, Nov. 1997.