

<http://doi.org/10.35784/iapgos.2066>

## DEVELOPMENT OF A MODULAR LIGHT-WEIGHT MANIPULATOR FOR HUMAN-ROBOT INTERACTION IN MEDICAL APPLICATIONS

Adam Kurnicki<sup>1</sup>, Bartłomiej Stańczyk<sup>2</sup>

<sup>1</sup>Lublin University of Technology, Automation and Metrology Department, Lublin, Poland, <sup>2</sup>Accrea Engineering, Lublin, Poland

**Abstract.** The article focuses on the design and implementation of mechanics, electronics and control system for a light-weight, modular, robotic manipulator for performing activities that require robot-human interaction in selected medicine-related applications. At the beginning, the functional requirements and physical architecture of such manipulator are discussed. The structure and control systems of the essential manipulator components/joint modules are presented in detail. Next, we introduce the software architecture of the master controller. Finally, examples of the current implementations of the modular manipulator are given.

**Keywords:** modular manipulator, physical architecture, control architecture

### OPRACOWANIE MODUŁOWEGO LEKKIEGO MANIPULATORA DO INTERAKCJI CZŁOWIEK-MASZYNA W ZASTOSOWANIACH MEDYCZNYCH

**Streszczenie.** Artykuł prezentuje zagadnienia związane z projektowaniem mechaniki, elektroniki i układów sterowania dla lekkiego, modułowego manipulatora robotycznego dedykowanego do wykonywania czynności wymagających interakcji człowiek-robot w wybranych aplikacjach medycznych. W pierwszej części artykułu omówiono wymagania funkcjonalne i architekturę fizyczną manipulatora. Następnie przedstawiono strukturę i układy sterowania podstawowych elementów manipulatora – modułów napędowych przegubów. Zaprezentowano architekturę oprogramowania sterowania implementowaną w sterowniku nadrzędnym. Na koniec podano przykłady zrealizowanych implementacji opracowanego manipulatora modułowego.

**Słowa kluczowe:** manipulator modułowy, architektura fizyczna, architektura oprogramowania

### Introduction

The rise and popularization of robot technology has already led to significant transformation in many fields of science and technology. In the 21<sup>st</sup> century, robots have not just played a significant role in industrial fields; their applications have also expanded to non-industrial fields. Various service and entertainment robots have entered family homes. Some are developed for medical or medicine-related applications in: surgery, rehabilitation, telediagnosics and to support mobility-impaired persons. They gradually become an important part of peoples' daily lives.

Most of the medical or medicine-related robotic systems utilize a wide variety of manipulators designed for physical human-robot interactions (PHRI). A general diagram of a PHRI system is presented in Fig. 1. An example of such a system is a teleoperation system where during telesurgery [5] or a remote medical examination [7], the user (doctor) generates commands for the manipulator via an input device (usually a haptic interface). Additionally, the doctor can feel the contact forces during the interaction with the environment, since the forces sensed at the manipulator end-effector are conveyed to his hand via a haptic interface. A display allows the user to observe the manipulation environment and to check the robot's operational status/mode. All measurement, control and vision signals are transmitted, usually over a long distance, via the Internet.

Another typical case of a PHRI system is an application in which the manipulator is used by persons with severe physical disabilities. This solution is dedicated especially for people who have no upper-limb or its usage is strongly limited. The

manipulator is usually integrated with a powered wheelchair [1, 8] and helps them in performing activities of daily living such as picking up and moving objects, eating and drinking, opening doors and switching lights and their TV on/off. In such a case, the user generates tasks for the manipulator with the use of a wheelchair joystick system, sometimes supported by additional specialized interfaces (e.g. sip-and-puff, body-machine interface [6]). Since the manipulator is placed close to the user, he/she can observe the arm movement directly and there is no need to transmit vision and control signals via the Internet. The user display, which is connected directly (by cable or Wi-Fi) to the manipulator controller, shows the actual arm state/mode. For users who are functionally locked-in due to any of a variety of neurological or physical conditions, instead of a classical input device, a brain computer interface [10] can be used.

From a design point of view, in order to be more commercially successful, the weight of the manipulator must be reduced while supporting a similar or increased payload, and the price should be decreased in comparison to available solutions. Reducing the weight of the manipulator will reduce the power consumption (e.g. allowing longer usage of the wheelchair batteries) and will increase the user safety. A lighter arm will also be less restrictive on the allowable user weight as specified by the wheelchair manufacturer. In order to achieve this, lightweight robots are normally designed using two approaches [3]. One approach is to design cable-driven robots by the allocation of motors in the base and transmission of their motions to the joints by tendon-like mechanisms, e.g. the design of the Barrett WAM arm or the Igus arm presented in [10]. This type of design leads to highly dexterous, naturally backdrivable and compliant actuations.

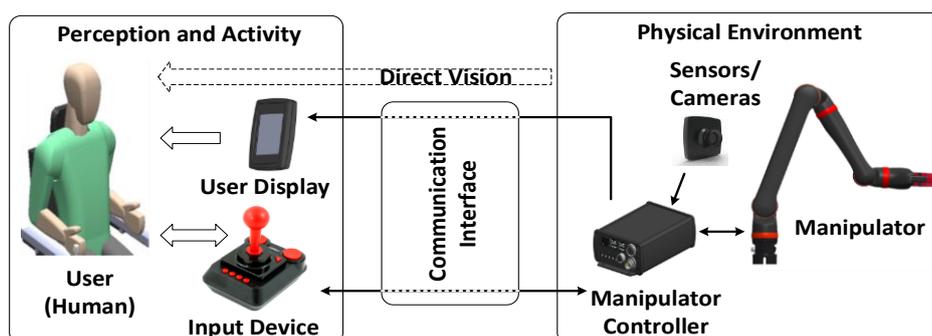


Fig. 1. General diagram of systems for physical human-robot interaction

The disadvantage of such solution is their large footprint with lower interchangeability. The other approach is to design with highly integrated components and to make the major structural components out of more technologically advanced materials such as composite materials [1, 8, 11]. The disadvantage of this solution is the higher cost of these materials. An advantage is the possibility to build modular (with better interchangeability), highly reliable and much more compact manipulators.

The goal of this project was to design a reliable, safe modular manipulator which is more cost-effective and compact, and has a greater or equal payload-to-weight ratio than the manipulators available on the market. The benefits of modularization [4], such as:

- versatility – using a few identical or different modules, various robots with different functionalities can be built quickly,
- reconfigurability – the kinematic structure of a robot may be modified by changing the mechanical configuration of the modules in the arm,
- scalability – the number of degrees of freedom of the robot can be changed by adding or removing the joint modules to the system,

allow a quasi-universal system to be developed, which can be used not only for one specific application, but can be easily adapted (configured) to different medicine-related applications.

The designed modular system is quite complex and hence required a solid development methodology. In this project, the methodology used was a result of merging the user-centred design approach (ISO-13407) and the “Design methodology of mechatronic systems” (VDI 2206). As a consequence, at the initial stage of the project development, the physical architecture of the manipulator was designed. This architecture, together with the chosen system requirements, is presented in section 1.

## 1. Functional requirements and physical architecture

According to the aforementioned system development methodology, the first step, which precedes the design process, is specification of the functional requirements. The most important requirements are the following:

- The structure of the manipulator shall be modular and configurable with the use of a maximum of 7 rotational joints and a maximum of 3 types of drive modules,

- The arm shall be capable of lifting a minimum of a 1 kg payload with an approx. 1 m reach,
- The maximum weight of the arm shall not exceed 5 kg,
- The width of the joint/link (length and diameter of the module) shall not exceed 9 cm – compactness – important especially for a wheelchair arm,
- The drive modules shall be multiturn, independent and complete mechatronic systems with switchable control modes (e.g. position/velocity/torque) and configurable parameters (e.g. motion limits, sensors calibration coefficients, etc.) by a higher level controller,
- It shall be possible to integrate the arm with a power wheelchair: mechanics, controller (via an IOM module) and 24 VDC battery power supply,
- The arm shall allow for eating and drinking from a bottle or cup, opening and closing doors and cupboards, switching on/off standard household equipment,
- The behavior (movement) of the arm shall be commanded by: a simple on/off joystick interface (implementation of standard control modes which allows: plane and up/down movement to be performed or to open/close and change the orientation of the gripper), 6DoF joystick (e.g. SpaceMouse) and external PC-based controller equipped with ROS (Robot Operating System) [13],
- The arm should operate only outside of a configurable *No Go Zone* and with limited speed in its definable vicinity (*Safety Zone* with configurable width),
- Additional safety features [2] shall be implemented to prevent non-controlled motion of the arm (e.g. self-check, failure detection and handling).

Based on the requirements analysis, the physical architecture presented in Fig 2 was designed. It has the form of a high-level diagram, where the whole system is split into two main physical components: *Arm* (manipulator) and *Master Controller* with their input/output interfaces. Both components can be supplied with voltage between 19 and 30 VDC, which is fully compatible with wheelchair batteries.

The *Arm* consists of: an aluminum base (with power and Ethernet sockets), up to seven joint modules connected by carbon fiber links (with power and Ethernet cables inside) and an end-effector (e.g. gripper). They are controlled by slave controllers run at a frequency of 1 kHz. The structure of the joint modules is described in section 2.

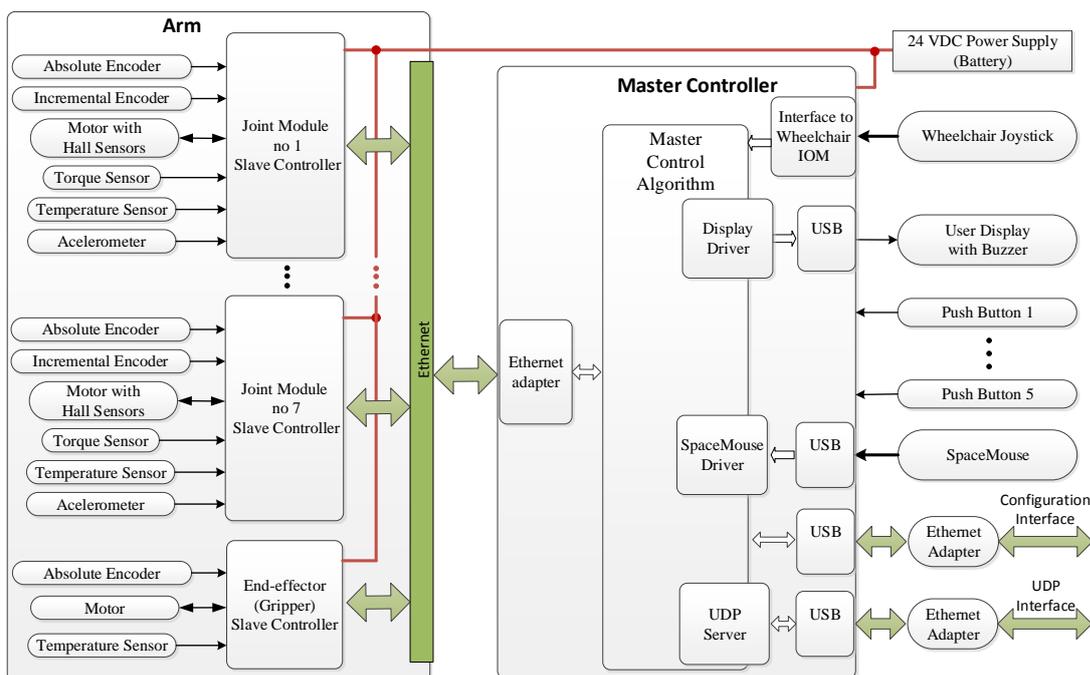


Fig. 2. Manipulator physical architecture

The *Master Controller* is based on a small, single-board Raspberry Pi 3 computer running real-time Linux. The real-time system allows for the execution of the master control algorithm at a frequency of 1 kHz. The software architecture of this algorithm is presented in section 3. The master control algorithm is supported by software drivers for the USB-connected external devices (user display, SpaceMouse) and by a UDP server which allows control-measurement data to be exchanged with an optional, higher-level control system based on a PC with a UDP client and, e.g., ROS. Push buttons and wheelchair controller signals are connected through a simple electronic logical interface to GPIOs on the Raspberry Pi.

Such an architecture and all of the above functional requirements imposed the implementation of a hierarchical two-level control system with a *Master Controller* at the higher-level and distributed slave controllers at the lower-level. The communication between the lower- and higher-level elements is performed at a frequency of 1 kHz via a real-time, Ethernet-based network.

A custom-designed PC application directly configures (through an Ethernet adapter connected to a USB port) the *Master Controller* and particular slave controllers.

## 2. Joint modules

Based on the manipulator requirements and the architecture presented in section 1, joint drive modules with two sizes and a three fingered gripper were designed. Each of them has integrated mechanics, electronics and control circuitry in one independent mechatronic system. A schematic diagram of a joint drive module is shown in Fig. 3.

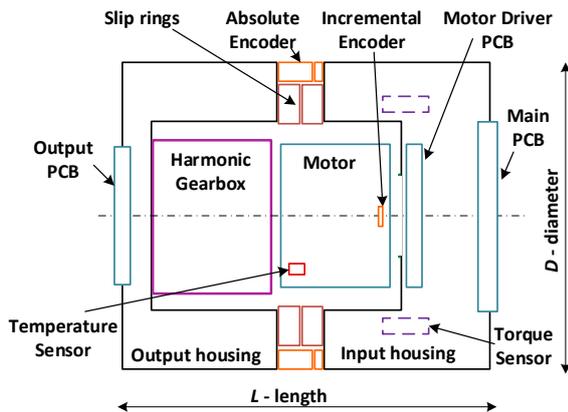


Fig. 3. Schematic diagram of the joint drive module

The structure of the drive module is based on two cylindrical load-bearing housings made of aluminum. They are mechanically connected to the drive system which consists of a brushless DC motor and a harmonic gearbox. Since the arm must be capable of lifting a 1 kg weight, two module sizes L – large and S – small (see Fig. 4) have been designed. They differ in the nominal output torque and the type of the components used. Their main parameters are collected in Table 1. The L-sized modules are normally mounted at the beginning of a serial manipulator kinematic chain and the S-sized at the end.

Table 1. Main parameters of joint modules

Module size	D [mm]	L [mm]	Nominal output torque [Nm]
L	75	71	12
S	62	55	5

Each module also contains: an incremental encoder mounted between the rotor and the stator of the motor, an absolute encoder mounted between the input and the output housing, a temperature sensor fixed at the stator windings and a strain gauge-based torque sensor glued onto the housing. In order to fulfill the multiturn requirement, custom-designed slip rings are used. They transmit the power and Ethernet signals between the input and output sides of the module.

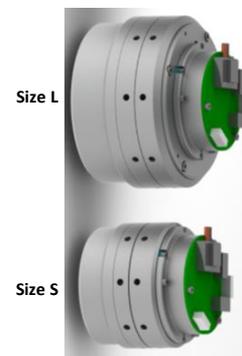


Fig. 4. Overview of L and S sizes of joint drive modules

The module electronics, shown in the block diagram in Fig. 5, was split across three PCBs. The first one, connected directly to the motor stator, is called the Motor Driver PCB and contains the hall sensors and the motor power stage with a driver. The second one, fixed at the left side of the output housing, is called the Output PCB and contains the power and Ethernet sockets. The last one, called the Main PCB, is mounted at right side of the input housing and supports the slave control algorithm implemented in a microcontroller.

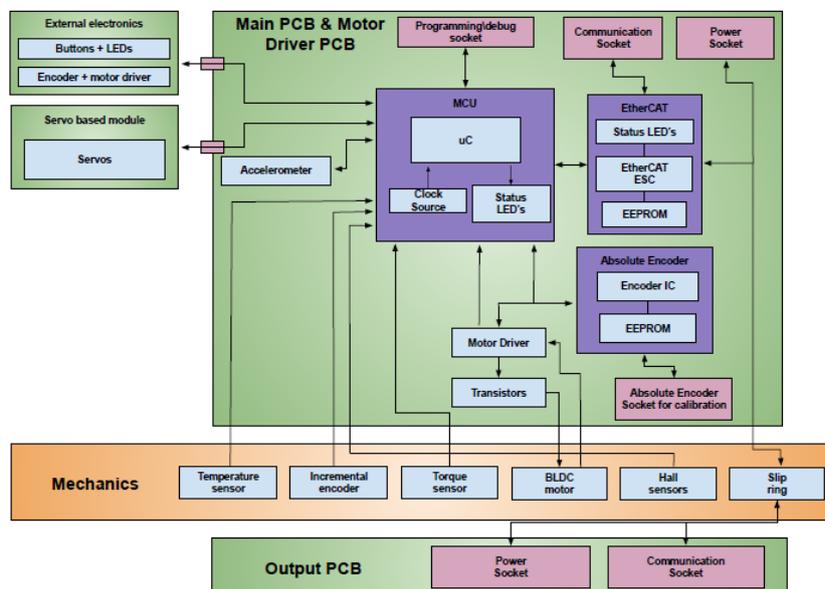


Fig. 5. Electronics block diagram of the joint module

The microcontroller peripherals are used to acquire the measurement data from the sensors (i.e. the accelerometer, the temperature sensor, the absolute and incremental encoders, the hall sensors and the torque sensor) and to generate commands and PWM signals for the motor power stage driver. They also communicate with the slave controller by handling the physical layer of the communication (i.e. processing of the frames exchanged with the manipulator master controller).

In the software layer, the communication is handled by the *Slave Ethernet Stack*. Through it, commands (demands) and configuration parameters are received and actual slave controller statuses are sent from/to the master controller. The *Slave Ethernet Stack* is part of the slave control algorithm, the architecture of which is presented in Fig. 6.

All input signals acquired from the sensors (hardware inputs) as well as demands from the master controller are validated and/or filtered and scaled in the *Data Acquisition & Signal Conditioning* block. For safety reasons (module self-check) these signals are analyzed and compared with thresholds, with each other or with modelled signals in the *Module Monitoring* block. If an anomaly is detected or if a fault is reported by the hardware, the block generates information about the critical or non-critical module failure, or only a warning. Based on the self-check result (monitoring status) and control commands received from the master controller, the *Decision Maker* manages the slave controller activities. The *Decision Maker*, which was designed in the form of a multilayered state machine [7], generates information about its current state (i.e. controller status), manages the *Cascade Controller* (enables/disables, switches working modes, etc.), switches the *Control Modes* of the slave controller (e.g. between: position, velocity, torque and failure handling control modes) and selects a proper filter structure for the demanded joint position signal.

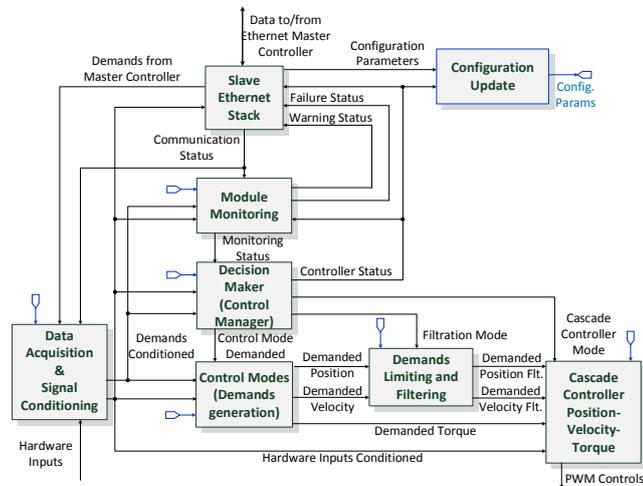


Fig. 6. The software architecture of the joint slave controller

The *Cascade Controller* has a standard position-velocity-torque cascade control structure presented with details in [7].

The *Demands Limiting and Filtering* block consists of: a filter (used to smooth the demanded joint position) and a speed limiter which limits the demanded joint velocity to a maximum permitted velocity when the joint is away from its physical limit and to a small value (zero) when close to it.

### 3. Master controller software architecture

A simplified software architecture of the master controller is presented in Fig. 7. It is an evolution of our previous work [7].

In general, it is used to generate the position  $q_{sd}$ , velocity  $\dot{q}_{sd}$  and torque  $\tau_{sd}$  demanded values for the slave controllers. These demanded values can be generated (by switchable applications of the *Joint Space Control Modes* block) and conditioned (by the *Joint Demands Filter with Speed Limiter, Integrator and Collision*

*Handling* block) directly in joint space, based on the position  $q_{UDPd}$ , velocity  $\dot{q}_{UDPd}$  or torque  $\tau_{UDPd}$  demanded values received from the external interface. However in most cases, generation of  $q_{sd}$  or  $\dot{q}_{sd}$  signals for slave controllers is more complex. First, the demanded task space velocity vector  $\dot{X}_d = [\dot{x}_d, \dot{\omega}_d]$  or demanded pose  $\xi_d$  is generated (by switchable applications of the *Task Space Control Modes* block) in task space, based on: wheelchair joystick signals  $W_{jd}$  or SpaceMouse joystick signals  $S_{jd}$  or a demanded pose  $\xi_{UDPd}$  received from the external interface. These demanded signals are partially conditioned (by the *Task Space Demands Filter and Admittance Control* block). Next, values of demanded joints velocities  $\dot{q}_{sd}$  are calculated by the *IK* (*Inverse Kinematics*) algorithm from demanded task velocity vector  $\dot{X}_{sd}$ . Finally, demanded joints velocities are limited and integrated (by *Speed Limiter and Integrator*) to obtain values of  $q_{sd}$  signals. The *IK* algorithm utilizes the velocity-based inverse kinematics algorithm [7]:

$$\dot{q}_{sd} = J_s^\# \cdot \dot{X}_{sd}, \quad (1)$$

where:  $J_s^\#$  is a Moore–Penrose pseudoinverse of the arm's Jacobian  $J_s$ . It is a solution designed to minimize the quadratic cost function of the joint velocities (according to the least square method [9]). In order to avoid the least square inverse method's problems with singularities, the weighted damped least square (WDLS) method was introduced for the manipulator IK algorithm as a modification of the DLS method [9]. Then minimizing the cost function:

$$g(\dot{q}_{sd}, \dot{X}_{sd}) = \frac{1}{2} (\dot{X}_{sd} - J_s \dot{q}_{sd})^T W_x (\dot{X}_{sd} - J_s \dot{q}_{sd}) + \frac{1}{2} \dot{q}_{sd}^T W_q \dot{q}_{sd}, \quad (2)$$

where:  $W_x$  and  $W_q$  are symmetric positive-definite weighting matrices associated with the errors in the task space and joint space, respectively, giving the following solution:

$$\dot{q}_{sd} = (J_s^T W_x J_s + W_q)^\# J_s^T W_x \dot{X}_{sd}. \quad (3)$$

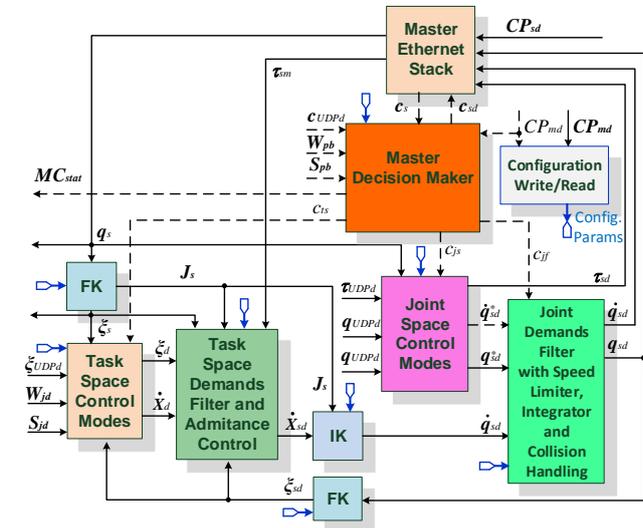


Fig. 7. Master controller software architecture

There is one additional, very important block, which was designed in the form of a finite state machine and manages the whole system – the *Master Decision Maker* (*MDM*). The *MDM*, based on slave controller statuses  $c_s$  and control commands received from the external devices (i.e. from: external interface –  $c_{UPDd}$ , wheelchair buttons –  $W_{pb}$  and SpaceMouse buttons –  $S_{pb}$ ), generates the demanded control modes for the slave controllers  $c_{sd}$  and commands:  $c_{js}$  and  $c_{js}$  which switch the task space, and joint space control modes of the master controller. The *MDM* reports its current state by signal  $MC_{stat}$ .

The configuration parameters of particular master controller's blocks can be updated on demand (triggered by the high state of the  $CP_{md}$  signal), with values coded in the  $CP_{md}$  signal. Both

signals used in the configuration process are received from externally connected (through Ethernet) PC application. A  $CP_{sd}$  signal, received from the same source, is used for the configuration of the slave controllers.

#### 4. Conclusion

The joint modules and the master controller presented in this article have already been implemented in two real medicine-related applications. The first solution is a modular, easy-to-reconfigure, light-weight manipulator mounted on wheelchair. This manipulator, shown in Fig. 8, helps to cope with physical disability in everyday life.



Fig. 8. Modular manipulator on a wheelchair

The second type of the manipulator was developed in order to replace a very heavy and user-unfriendly arm of the RAMCIP (Robotic Assistant for MCI Patients at home) robot [12]. The RAMCIP robot with our manipulator is shown in Fig. 9.

In both cases, custom-designed grippers were used, with electronics and control systems similar to the one dedicated to the joint modules presented in section 2. Preliminary evaluation results with real users have confirmed the operational correctness of both the manipulators and their control systems. Our future development related to the modular manipulator will concentrate on its implementation as an arm for remote ultrasound examination.



Fig. 9. Modular manipulator on a RAMCIP robot

#### Acknowledgements

The work presented in this paper was supported in part by the Lublin Enterprise Support Agency as part of the “Research and innovation” project, activity 1.2, RPO WL 2014-2020.

#### References

- [1] Campeau-Lecours A., et al.: Kinova Modular Robot Arms for Service Robotics Applications. *International Journal of Robotics Applications and Technologies* 5(2), 2017, 49–71 [http://doi.org/10.4018/IJRAT.2017070104].
- [2] Dibekci A., Bebek O.: Improving the Safety of Medical Robotic Systems. 7<sup>th</sup> IEEE International Conference on Biomedical Robotics and Biomechanics, 2018, 73–78. [http://doi.org/10.1109/BIOROB.2018.8487914].
- [3] Fang H., Guo L., Bai S.: A Light Weight Arm Designed with Modular Joints. Bai S., Ceccarelli M. (eds): *Recent Advances in Mechanism Design for Robotics*. *Mechanisms and Machine Science* 33, 2015, 47–54 [http://doi.org/10.1007/978-3-319-18126-4\_5].
- [4] Guan Y., Jiang L., Zhang X., Zhang H., Zhou X.: Development of novel robots with modular methodology. *IEEE/RSJ International Conference on Intelligent Robots and Systems* 2009, 2385–2390, [http://doi.org/10.1109/IROS.2009.5354051].
- [5] Hassan T., Hameed A., Nisar S., Kamal N., Hasan O., Al-Zahravi: A Telesurgical Robotic System for Minimal Invasive Surgery. *IEEE Systems Journal* 10(3), 2016, 1035–1045 [http://doi.org/10.1109/JSYST.2014.2331146].
- [6] Jain S., Farshchiansadegh A., Broad A., Abdollahi F., Mussa-Ivaldi F., Argall B.: Assistive robotic manipulation through shared autonomy and a Body-Machine Interface. *IEEE International Conference on Rehabilitation Robotics*, 2015, 526–531, [http://doi.org/10.1109/ICORR.2015.7281253].
- [7] Kurnicki A., Stańczyk B.: Manipulator Control System for Remote USG Examination. *Journal of Automation, Mobile Robotics and Intelligent Systems* 13(2), 2019, 48–59 [http://doi.org/10.5604/20830157.1121333].
- [8] Schrock P., Farello F., Alqasemi R., Dubey R.: Design, simulation and testing of a new modular wheelchair mounted robotic arm to perform activities of daily living. *IEEE International Conference on Rehabilitation Robotics*, 2009, 518–523, [http://doi.org/10.1109/ICORR.2009.5209469].
- [9] Siciliano B., Sciacivico L., Villani L., Oriolo G.: *Robotics. Modelling, Planning and Control*. *Advanced Textbooks in Control and Signal Processing*, Springer-Verlag, London 2009 [http://doi.org/10.1007/978-1-84628-642-1].
- [10] Tremblay T., Padir T.: Modular Robot Arm Design for Physical Human-Robot Interaction. *IEEE International Conference on Systems, Man, and Cybernetics*, 2013, 4482–4487, [http://doi.org/10.1109/SMC.2013.762].
- [11] Vogel J., Haddadin S., Simeral J. D., Stavisky S.D., Bacher D., Hochberg L. R., Donoghue J. P.: Continuous Control of the DLR Light-Weight Robot III by a Human with Tetraplegia Using the BrainGate2 Neural Interface System. Khatib O., Kumar V., Sukhatme G. (eds): *Experimental Robotics*. Springer Tracts in Advanced Robotics 79. Springer, Berlin, Heidelberg 2014 [http://doi.org/10.1007/978-3-642-28572-1\_9].
- [12] Ramcip project homepage, <https://ramcip-project.eu/> (accessed: 27.06.2020).
- [13] ROS Homepage, <https://www.ros.org/> (accessed: 27.06.2020).

#### Ph.D. Eng. Adam Kurnicki

e-mail: a.kurnicki@pollub.pl

Received his Ph.D. (2004) in technical science and his M.Sc. (1999) from the Faculty of Electrical Engineering and Computer Science, TU Lublin, Poland. He is a lecturer in the Automation and Metrology Department, TU Lublin, where he is involved in research on control systems dedicated mainly for robotic systems used in medical related applications.

<https://orcid.org/0000-0002-4988-7322>

#### Ph.D. Eng. Bartłomiej Stańczyk

e-mail: b.stanczyk@accrea.com

Received his Ph.D. (2006) in technical science from TU Muenchen, Germany and his M.Sc. in electrical and control engineering (1998) from TU Lublin, Poland. Between 1998 and 2006, worked as a research assistant at TU Lublin (Poland), Berlin and Munich (Germany), involved in various research projects on robust and digital control. During his Ph.D. period, he focused on teleoperation, redundant kinematics and impedance control of robotic manipulators. Since 2014, he is full time technical director of ACCREA.

<https://orcid.org/0000-0002-2319-7358>

otrzymano/received: 3.07.2020

przyjęto do druku/accepted: 10.12.2020

