ARCHIVE OF MECHANICAL ENGINEERING

Volume 67 2020 Number 1

DOI: 10.24425/ame.2020.131686

Key words: adaptive control, neural network control, redundant manipulator

Łukasz Woliński [1]

Comparison of the adaptive and neural network control for LWR 4+ manipulators: simulation study

This paper deals with two control algorithms which utilize learning of their models' parameters. An adaptive and artificial neural network control techniques are described and compared. Both control algorithms are implemented in MATLAB and Simulink environment, and they are used in the simulation of a postion control of the LWR 4+ manipulator subjected to unknown disturbances. The results, showing the better performance of the artificial neural network controller, are shown. Advantages and disadvantages of both controllers are discussed. The usefulness of the learning algorithms for the control of LWR 4+ robots is discussed. Preliminary experiments dealing with dynamic properties of the two LWR 4+ robots are reported.

1. Introduction

The problem of the robotic manipulator control stretches back to the early days of robotics. During the second half of the last century, many approaches – from the independent joint PID control to the model-based methods [1–4] – were developed. Based on the manipulator dynamic model, computed torque methods offer good tracking performance [1–4], however they obviously require the knowledge of the dynamic parameters (masses, locations of the centers of mass and moments of inertia) of the manipulator. The identification of such parameters is often a demanding task during which a researcher has to solve problems of selecting the set of identifiable parameters and obtaining a sufficiently rich data [5, 6]. Moreover, the mass and inertia of the object carried by the end-effector would also have to be identified. If the parameters are not known with the sufficient accuracy, the control quality might not be acceptable.

✉ Łukasz Woliński, email: lwolinski@meil.pw.edu.pl

[1] Institute of Aeronautics and Applied Mechanics, Warsaw University of Technology, Poland.

To overcome the aforementioned problems, an adaptive control scheme for robotic manipulators was proposed in the 1980s [7–11]. In this approach, the parameters of the model are estimated online, based on the position and velocity errors. Consequently, there is no need for the prior parameters identification. The model is constantly updating itself, ensuring small tracking error.

On the other hand, another approach – the artificial neural network control – does not use any dynamic model at all. Given that neural networks are universal approximators, they can represent unknown nonlinear manipulator dynamics [12, 13]. The use of neural networks in closed-loop feedback control systems was studied since at least the 1990s [12–18]. Similarily as in the case of adaptive control, a neural network controller constantly updates its weights to keep the tracking error in check.

As shown, the adaptive and neural network control are not entirely new concepts, however they are still actively researched and extended nowadays as evidenced in [19–26].

In this paper the adaptive and neural controllers are discussed, and implemented in the simulation of the closed-loop feedback control system. The controlled object is a dynamic model of the KUKA lightweight redundant robotic manipulator (LWR 4+). The primary objective of this work is the comparison of the two control methods.

The paper is organized as follows: section 2 deals with the controlled object, sections 3.1 and 3.2 describe the adaptive and artificial neural network controllers, respectively. In section 4 results of the simulation study are detailed, in section 5 preliminary works on the robots are discussed and section 6 concludes the paper.

2. Model of the controlled object

The studied controlled object is the KUKA LWR 4+ manipulator. LWR 4+ is a 7 degree of freedom light-weight robot with a redundant anthropomorphic structure. To develop the dynamic model of the manipulator, a recursive algorithm presented in [27] was used. It is based on a spatial operator algebra [28] and derives the equations of motion of the dynamic system in a compact matrix form.

The manipulator is modeled as a multibody system comprised of n rigid links and the equations are formulated in the joint space. For simplicity, the joint elasticity and actuator dynamics are not included in the model. The equations of motion of the manipulator are given in the matrix form:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \boldsymbol{\tau}, \tag{1}$$

where $\mathbf{q} \in \mathbb{R}^{n\times 1}$ is the vector of joint coordinates, $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{n\times n}$ is the manipulator inertia matrix, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} \in \mathbb{R}^{n\times 1}$ is the Coriolis and centrifugal force vector, $\mathbf{G}(\mathbf{q}) \in \mathbb{R}^{n\times 1}$ is the gravitational force vector, $\boldsymbol{\tau} \in \mathbb{R}^{n\times 1}$ contains driving torques in joints and $n = 7$ as the manipulator has 7 degrees of freedom.

The kinematic data necessary for the model derivation was obtained from the official KUKA documentation [29]. The dynamic data, such as inertias of the links, were obtained through the identification procedures described in [5, 6] and supplemented with the parameters published in [30].

3. Controllers

3.1. Adaptive controller

The considered adaptive controller [11, 31] minimizes the position error while estimating online the unknown dynamic parameters. The control law is given by the following equation:

$$\boldsymbol{\tau} = \mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r, \ddot{\mathbf{q}}_r)\hat{\boldsymbol{\theta}} + \mathbf{K}\mathbf{s}, \tag{2}$$

where $\mathbf{K} \in \mathbb{R}^{n\times n}$ is the positive definite gain matrix, and $\mathbf{s}$ is the filtered tracking error:

$$\mathbf{s} = \dot{\mathbf{q}}_r - \dot{\mathbf{q}}, \tag{3}$$

while $\dot{\mathbf{q}}_r$ is the reference joint velocity defined as:

$$\dot{\mathbf{q}}_r = \dot{\mathbf{q}}_d + \boldsymbol{\Lambda}\mathbf{e}. \tag{4}$$

The joint position error $\mathbf{e}$ is computed as:

$$\mathbf{e} = \mathbf{q}_d - \mathbf{q}, \tag{5}$$

where $\mathbf{q}_d$ is the desired joint trajectory and $\boldsymbol{\Lambda} \in \mathbb{R}^{n\times n}$ is a symmetric positive definite gain matrix. The current estimate of the dynamic parameters $\boldsymbol{\theta} \in \mathbb{R}^{p\times 1}$ is contained in the vector $\hat{\boldsymbol{\theta}} \in \mathbb{R}^{p\times 1}$, where p is the number of the parameters. The manipulator regressor matrix $\mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r, \ddot{\mathbf{q}}_r) \in \mathbb{R}^{n\times p}$ satisfies the equation [31]:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}}_r + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}_r + \mathbf{G}(\mathbf{q}) = \mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r, \ddot{\mathbf{q}}_r)\hat{\boldsymbol{\theta}}, \tag{6}$$

which is based on the linearity-in-parameters property of the rigid body dynamic manipulator model [3, 32].

Finally, the parameter adaptation law is given by:

$$\dot{\hat{\boldsymbol{\theta}}} = \boldsymbol{\Gamma}\mathbf{Y}^T(\mathbf{q}, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r, \ddot{\mathbf{q}}_r)\mathbf{s}, \tag{7}$$

where $\boldsymbol{\Gamma} \in \mathbb{R}^{p\times p}$ is a positive definite gain matrix.

The main difficulty of implementing that control scheme is the derivation of the regressor $\mathbf{Y}$. In general, this is a demanding task [3, 32]. However, an efficient algorithm, based on the recursive Newton-Euler formulation, for computing the regressor $\mathbf{Y}$ is presented in [31]. It requires only the knowledge of the kinematic parameters such as link lenghts and locations of the joints. The stability issues of the adaptive control method are discussed in [11, 31]. Number of the dynamic parameters p (and therefore the number of columns of $\mathbf{Y}$) can be reduced to account only for the most essential parameters in the model using the algorithm described in [33].

Nevertheless, to parametrize the model in the unknown parameters, the structure of the manipulator has to be fully known. That is, the model has to account for all the significant dynamics. Furthermore, if the manipulator dynamics includes considerable nonlinear components (e.g. some forms of friction) then the linearity-in-parameters property does not hold. In turn, the control quality might deteriorate due to the presence of the unmodeled dynamics as the discussed controller relies on the linear model.

Moreover, the elements of $\hat{\boldsymbol{\theta}}$ does not necessarily converge to the real values of the dynamic parameters given in $\boldsymbol{\theta}$. Similarly as in the case of the offline identification, a sufficiently exciting trajectory has to be used as the desired trajectory $\mathbf{q}_d$ to obtain the real $\boldsymbol{\theta}$ [11].

3.2. Neural network controller

As mentioned in the introduction, the artificial neural network (ANN) controller does not use a dynamic model. Instead, the neural network compensates the unknown nonlinear dynamics. No reliance on the linearity-in-parameters assumption makes the use of ANN controller advantageous over the adpative controller.

The considered controller is based on [12, 17]. It is a closed-loop feedback system containing a two-layer ANN. The first layer – also called the input layer – has l neurons, while the second – or the hidden – layer has h neurons. The equation describing the control law is given below:

$$\boldsymbol{\tau} = \mathbf{W}^T \boldsymbol{\sigma}_a + \mathbf{K}\mathbf{s}, \tag{8}$$

where $\boldsymbol{\sigma}_a \in \mathbb{R}^{(h+1)\times 1}$ is defined as:

$$\boldsymbol{\sigma}_a = \begin{bmatrix} 1 \\ \sigma\left(\mathbf{V}^T \mathbf{X}\right) \end{bmatrix}, \tag{9}$$

and where $\mathbf{X} \in \mathbb{R}^{l+1}$:

$$\mathbf{X} = \begin{bmatrix} 1 & \mathbf{e}^T & \dot{\mathbf{e}}^T & \mathbf{q}_d^T & \dot{\mathbf{q}}_d^T & \ddot{\mathbf{q}}_d^T \end{bmatrix}^T \tag{10}$$

is the augmented vector of the inputs to the neural network (the first element equal to 1 allows to include neurons' biases), $\mathbf{V} \in \mathbb{R}^{(l+1)\times h}$ is the matrix of biases (the first row) and weights (the rest l rows) of the input layer, and $\mathbf{W} \in \mathbb{R}^{(h+1)\times n}$ is the matrix of biases (the first row) and weights (the rest h rows) of the hidden layer.

Each element of the vector $\boldsymbol{\sigma}$ is a sigmoid activation function given by:

$$\sigma_j = \frac{1}{1+\exp(-z_j)}, \quad j = 1,2,\ldots,h, \tag{11}$$

where z_j are the outputs of the input layer. In other words, z_j are the elements of the vector $\mathbf{z} = \mathbf{V}^T\mathbf{X}$.

Standard open-loop weight training algorithms do not apply to the closed-loop feedback systems where weights as well as all internal states have to be bounded and the tracking error has to remain small [12, 17, 18]. One of the solutions is to use the backpropagation tuning algorithm, for example, the augmented backprop tuning [12]:

$$\dot{\mathbf{W}} = \mathbf{F}\boldsymbol{\sigma}_a\mathbf{s}^T - \mathbf{F}\boldsymbol{\sigma}'_a\mathbf{V}^T\mathbf{X}\mathbf{s}^T - k_w\mathbf{F}||\mathbf{s}||\mathbf{W}, \tag{12}$$

$$\dot{\mathbf{V}} = \mathbf{B}\mathbf{X}(\boldsymbol{\sigma}'^T_a\mathbf{W}\mathbf{s})^T - k_w\mathbf{B}||\mathbf{s}||\mathbf{V}, \tag{13}$$

where the design parameters $\mathbf{F} \in \mathbb{R}^{(h+1)\times(h+1)}$ and $\mathbf{B} \in \mathbb{R}^{(l+1)\times(l+1)}$ are positive definite matrices and k_w is a positive scalar, while $||\mathbf{x}||$ is the Euclidean norm of $\mathbf{x}$. Finally, $\boldsymbol{\sigma}'_a \in \mathbb{R}^{(h+1)\times h}$ is a derivative of $\boldsymbol{\sigma}_a$, defined as:

$$\boldsymbol{\sigma}'_a = \begin{bmatrix} 0 & 0 & 0 & \ldots & 0 \\ \sigma'_1 & 0 & 0 & \ldots & 0 \\ 0 & \sigma'_2 & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \ldots & \vdots \\ 0 & 0 & 0 & \ldots & \sigma'_h \end{bmatrix}, \tag{14}$$

where:

$$\sigma'_j = \sigma_j\left(1-\sigma_j\right), \quad j = 1,2,\ldots,h, \tag{15}$$

and σ_j is given by (11).

The stability of the ANN controller is covered in detail in [12, 18].

4. Simulations and results

The simulations were performed in the MATLAB/Simulink. The equation (1) was extended by adding the vector $\boldsymbol{\tau}_{dist} \in \mathbb{R}^{n\times 1}$ representing unknown nonlinear dynamics – such as friction in joints – and other disturbances:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) + \boldsymbol{\tau}_{dist} = \boldsymbol{\tau}. \tag{16}$$

It was chosen arbitrarily as:

$$\boldsymbol{\tau}_{dist} = \begin{bmatrix} 0.5\sin\left(0.1t + \dfrac{\pi}{6}\right) \\ 1.5\sin\left(0.25t + \dfrac{\pi}{10}\right) \\ -\sin(0.2t) \\ 0.5\sin(0.15t) \\ -0.2\sin(0.1t) \\ 0.1\sin(0.15t) \\ 0.01\sin(0.1t) \end{bmatrix} \text{Nm}. \tag{17}$$

The end-effector trajectory was designed as a circle with the radius of 0.3 m to be completed in 10 seconds:

$$\mathbf{r}_d(t) = \begin{bmatrix} 0.276 \\ -0.3 + 0.3\cos\left(\dfrac{\pi}{5}t\right) \\ 0.124 + 0.3\sin\left(\dfrac{\pi}{5}t\right) \end{bmatrix} \text{m} \tag{18}$$

and the joint trajectories were calculated by solving the inverse kinematics problem and using the manipulability measure to avoid singularities [32]:

$$\dot{\mathbf{q}} = \mathbf{J}^{+}\dot{\mathbf{r}}_d + \left(\mathbf{I} - \mathbf{J}^{+}\mathbf{J}\right) k_q \nabla H(\mathbf{q}), \tag{19}$$

where $\mathbf{J}^{+}$ is the pseudoinverse of the manipulator's Jacobian matrix $\mathbf{J}(\mathbf{q}) \in \mathbb{R}^{3\times7}$, k_q is the scalar gain and $H(\mathbf{q})$ is the manipulability measure defined as:

$$H(q) = \sqrt{\det\left(\mathbf{J}\mathbf{J}^{T}\right)}. \tag{20}$$

The driving torques in the vector $\boldsymbol{\tau}$ at the right hand side of the Eq. (16) were computed using the Eqs. (2) and (8). The adaptive controller was implemented with $p = 8$ dynamic parameters and $\boldsymbol{\Gamma} = 200\mathbf{I}_{8\times8}$. Meanwhile, the parameters of the ANN controller were chosen as follows: number of neurons in the hidden layer $h = 7$, number of inputs to the neural network $l = 35$, $\mathbf{F} = 200\mathbf{I}_{8\times8}$, $\mathbf{B} = 200\mathbf{I}_{36\times36}$, and $k_w = 0.5$. Both controllers used $\mathbf{K} = \text{diag}([10\ 100\ 60\ 80\ 10\ 10\ 10])$ and $\boldsymbol{\Lambda} = 5\mathbf{I}_{7\times7}$. All the dynamic parameters in the adaptive controller and weights in the ANN controller were initialized to zero.

Figs. 1, 2 and 3 show the end-effector position errors for the adaptive and ANN controller.

It is evident that the ANN controller performs better than the adaptive one, except for the z-axis where the gravity forces are dominating the unknown disturbances. These results illustrate the inability of the adaptive controller to mitigate

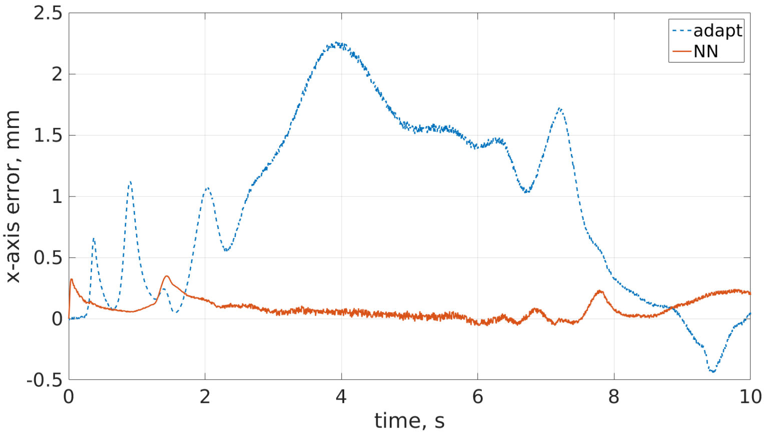

Fig. 1. X-axis position error

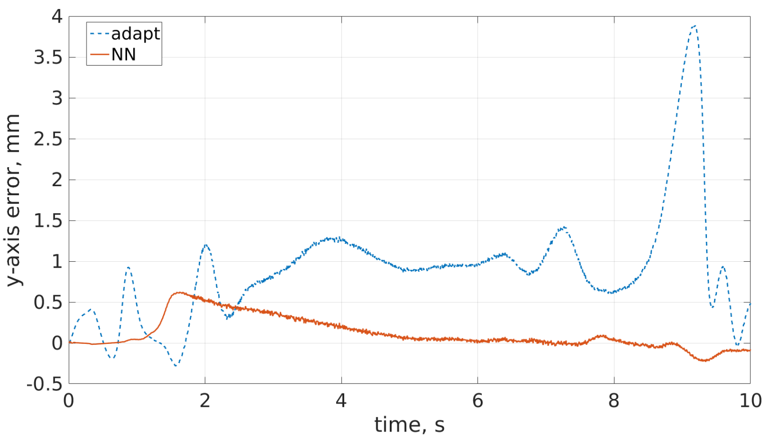

Fig. 2. Y-axis position error

the effects not accounted in the linear model of Eq. (6). Meanwhile, the ANN managed to achieve good level of position tracking despite the disturbances, as it was expected.

On the other hand, the ANN requires a lot of design parameters (matrices **F** and **B**) which are subject to heuristic tuning. Additionaly, the number of weights of the ANN:

$$N_{weights} = (l+1)h + (h+1)n = 252 + 56 = 308 \tag{21}$$

is much greater than of the parameters of the dynamic model ($p = 8$). Moreover, the dynamic parameters of the model-based adaptive controller have a physical interpretation like masses or moments of inertia which can not be said about the ANN weights.

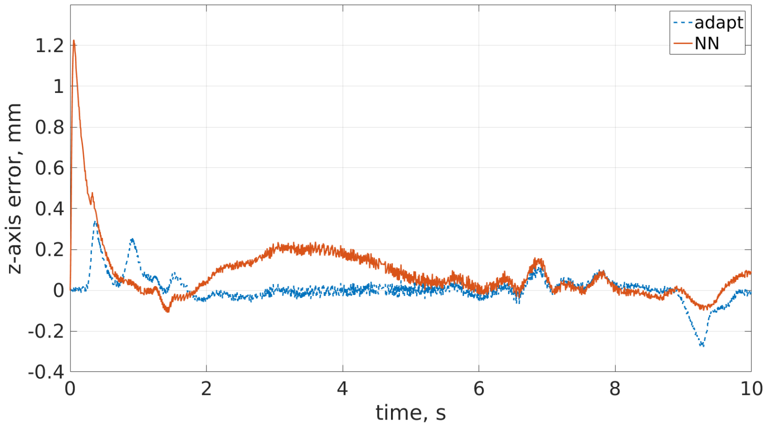

Fig. 3. Z-axis position error

5. Preliminary work on the robots

Preliminary experiments carried on two KUKA LWR 4+ robots showed subtle but apparent differences in the dynamic properties of the manipulators [34]. The experiments consisted in programming both robots to follow the same sinusoidal trajectories, measuring the joint torques and comparing the results with the rigid-body model given by Eq. (1). It was observed that the differences between the two robots and between the dynamic model and each robot were comparable in magnitude. These results suggested that the unmodeled effects such as friction or joint elasticity are possible culprits. It was concluded that in order to account for that phenomena in the model, each robot would have to be identified separately. On the other hand, if such level of detail is not necessary for a given application, a rigid-body model, obtained from identifying just one robot, should suffice. Alternatively, for control purposes, an adaptive or ANN controller studied in this paper could be used. Possibly, the parameters and weights would converge to slightly different values when used on two LWR 4+ robots doing the same tasks.

6. Conclusions

In this paper, an adaptive and ANN control methods were described and their features were compared. Both approaches were implemented in MATLAB/Simulink and used in the simulation of a postion control of the LWR 4+ manipulator subjected to unknown disturbances. The results show the advantage of the ANN over the adaptive control method. However, the model-based adaptive controller still has its advantages. Future works will include formulating the model in the canonical coordinates, as the Hamilton formulation has many advantages in the control systems [35–37].

Moreover, the preliminary experiments carried on two LWR 4+ robots were discussed. It was discovered that there are subtle differences in the dynamic properties of the two robots. Given that, the learning control algorithm such as the adaptive controller or the ANN is clearly advantageous over the non-learning model-based controller which would require the separate identification of each robot. In that regard, an implementation of the ANN algorithm on the real LWR 4+ is planned with the use of the KUKA FRI programming interface, which allows to send the desired joint torques from the user's computer to the robot's controller in each control cycle [38].

Acknowledgements

This work was supported by the National Science Centre (Poland) grant no. 2018/29/B/ST8/00374.

Manuscript received by Editorial Board, January 24, 2020;
final version, April 02, 2020.

References

[1] J. Craig. *Introduction to Robotics. Mechanics & Control*. Addison-Wesley Publishing Company, 1986.

[2] R. Kelly, V.S. Davila, and A. Loría. *Control of Robot Manipulators in Joint Space*. Springer, London, 2005. doi: 10.1007/b135572.

[3] M.W. Spong, S. Hutchinson, and M. Vidyasagar. *Robot Modeling and Control*. John Wiley & Sons, 2006.

[4] F.W. Lewis, D.M. Dawson, and C.T. Abdallah. *Robot Manipulator Control: Theory and Practice*. CRC Press, 2003.

[5] J. Swevers, C. Ganseman, D.B.Tukel, J. de Schutter, and H. Van Brussel. Optimal robot excitation and identification. *IEEE Transactions on Robotics and Automation*, 13(5):730–740, 1997. doi: 10.1109/70.631234.

[6] J.Swevers, W. Verdonck, and J. de Schutter. Dynamic model identification for industrial robots. *IEEE Control Systems Magazine*, 27(5):58–71, 2007. doi: 10.1109/MCS.2007.904659.

[7] A. Liegeois, E. Dombre, and P. Borrel. Learning and control for a compliant computer-controlled manipulator. *IEEE Transactions on Automatic Control*, 25(6):1097–1102, 1980. doi: 10.1109/TAC.1980.1102513.

[8] A.J. Koivo and T.H. Guo. Control of robotic manipulator with adaptive controller. In *1981 20th IEEE Conference on Decision and Control including the Symposium on Adaptive Processes*, pages 271–276, San Diego, USA, 16–18 Dec. 1981. doi: 10.1109/CDC.1981.269527.

[9] C.S.G. Lee and M.J. Chung. An adaptive control strategy for computer-based manipulators. In *1982 21st IEEE Conference on Decision and Control*, pages 95–100, Orlando, USA, 8–10 Dec. 1982. doi: 10.1109/CDC.1982.268407.

[10] A. Koivo and T.H. Guo. Adaptive linear controller for robotic manipulators. *IEEE Transactions on Automatic Control*, 28(2):162–171, 1983. doi: 10.1109/TAC.1983.1103211.

[11] J.-J.E. Slotine and W. Li. On the adaptive control of robot manipulators. *The International Journal of Robotics Research*, 6(3):49–59, 1987. doi: 10.1177/027836498700600303.

[12] F.W. Lewis, S. Jagannathan, and A. Yesildirak. *Neural Network Control ff Robot Manipulators and Non-Linear Systems*. Taylor & Francis, Inc, 1998.

[13] G. Dreyfus, G. *Neural Networks. Methodology and Applications*. Springer-Verlag, Berlin, Heidelberg, 2005. doi: 10.1007/3-540-28847-3.

[14] M.A. Johnson and M.B. Leahy. Adaptive model-based neural network control. *IEEE International Conference on Robotics and Automation Proceedings*, volume 3, pages 1704-1709, Cincinnati, USA, 13–18 May 1990. doi: 10.1109/ROBOT.1990.126255.

[15] M.B. Leahy, M A. Johnson, D.E. Bossert, and G.B. Lamont. Robust model-based neural network control. In *1990 IEEE International Conference on Systems Engineering*, pages 343–346, Pittsburgh, USA, 9–11 Aug. 1990. doi: 10.1109/ICSYSE.1990.203167.

[16] R.T. Newton and Y. Xu. Neural network control of a space manipulator. *IEEE Control Systems Magazine*, 13(6):14–22, 1993. doi: 10.1109/37.247999.

[17] F.L. Lewis. Neural network control of robot manipulators. *IEEE Expert*, 11(3):64–75, 1996. doi: 10.1109/64.506755.

[18] F.L. Lewis, A. Yesildirek, and K. Liu. Multilayer neural-net robot controller with guaranteed tracking performance. *IEEE Transactions on Neural Networks*, 7(2):388–399, 1996. doi: 10.1109/72.485674.

[19] A. Bottero, G. Gerio, V. Perna, and A. Gagliano. Adaptive control techniques and feed forward compensation of periodic disturbances in industrial manipulators. In *2014 IEEE/ASME 10th International Conference on Mechatronic and Embedded Systems and Applications (MESA)*, pages 1–7, Senigallia, Italy, 10–12 Sept. Sept. 2014. doi: 10.1109/MESA.2014.6935612.

[20] J. Li, H. Ma, C. Yang, and M. Fu. Discrete-time adaptive control of robot manipulator with payload uncertainties. In *2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, pages 1971–1976, Shenyang, China, 8–12 June 2015. doi: 10.1109/CYBER.2015.7288249.

[21] M. Li, Y. Li, S.S. Ge, and T.H. Lee. Adaptive control of robotic manipulators with unified motion constraints. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(1):184–194, 2017. doi: 10.1109/TSMC.2016.2608969.

[22] Ł. Woliński. Implementation of the adaptive control algorithm for the KUKA LWR 4+rRobot. In J. Awrejcewicz, ed., *Dynamical Systems in Theoretical Perspective*, volume 248 of Springer Proceedings in Mathematics & Statistics, pages 391–401, Springer, Cham, 2018. doi: 10.1007/978-3-319-96598-7_31.

[23] M. de Paula Assis Fonseca, B.V. Adorno, and P. Fraisse. An adaptive controller with guarantee of better conditioning of the robot manipulator joint-space inertia matrix. In *2019 19th International Conference on Advanced Robotics (ICAR)*, pages 111–116, Belo Horizonte, Brazil, 2–6 Dec. 2019. doi: 10.1109/ICAR46387.2019.8981558.

[24] L. Zhang and L. Cheng. An adaptive neural network control method for robotic manipulators trajectory tracking. In *2019 Chinese Control And Decision Conference (CCDC)*, pages 4839–4844, Nanchang, China, 3–5 June 2019. doi: 10.1109/CCDC.2019.8832715.

[25] He Jun-Pei, Huo Qi, Li Yan-Hui, Wang Kai, Zhu Ming-Chao, and Xu Zhen-Bang. Neural network control of space manipulator based on dynamic model and disturbance observer. *IEEE Access*, 7:130101–130112, 2019. doi: 10.1109/ACCESS.2019.2937908.

[26] A. Nawrocka, M. Nawrocki, and A. Kot. Neural network control for robot manipulator. In *2019 20th International Carpathian Control Conference (ICCC)*, pages 1–4, Krakow-Wieliczka, Poland, 26–29 May 2019. doi: 10.1109/CarpathianCC.2019.8765995.

[27] Ł. Woliński and P. Malczyk. Dynamic modeling and analysis of a lightweight robotic manipulator in joint space. *Archive of Mechanical Engineering*, 62(2):279–302, 2015. doi: 10.1515/meceng-2015-0016.

[28] G. Rodriguez, A. Jain, and K. Kreutz-Delgado. A spatial operator algebra for manipulator modelling and control. *International Journal of Robotics Research*, 10(4):371–381, 1991. doi: 10.1177/027836499101000406.

[29] *Lightweight Robot 4+ Specification, Version: Spez LBR 4+ V2en*, 06.07.2010.

[30] A. Jubien, M. Gautier, and A. Janot. Dynamic identification of the Kuka lightweight robot: comparison between actual and confidential Kuka's parameters. In *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics 2014*, pages 483–488, Besancon, France, 8-11 July 2014. doi: 10.1109/AIM.2014.6878124.

[31] H. Kawasaki, T. Bito, and K. Kanzaki. An efficient algorithm for the model-based adaptive control of robotic manipulators. *IEEE Transactions on Robotics and Automation*, 12(3):496–501, 1996. doi: 10.1109/70.499832.

[32] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo. *Robotics. Modelling, Planning and Control*. Springer-Verlag, London, 2009. doi: 10.1007/978-1-84628-642-1.

[33] M. Gautier and W. Khalil. Direct calculation of minimum set of inertial parameters of serial robots. *IEEE Transactions on Robotics and Automation*, 6(3):368-373, 1990. doi: 10.1109/70.56655.

[34] Ł. Woliński and M. Wojtyra. Comparison of dynamic properties of two KUKA lightweight robots. In *ROMANSY 21 – Robot Design, Dynamics and Control. Proceedings of the 21st CISM-IFToMM Symposium*, volume 569, pages 413–420, 2016. doi: 10.1007/978-3-319-33714-2_46.

[35] V. Záda and K. Belda. Mathematical modeling of industrial robots based on Hamiltonian mechanics. In *2016 17th International Carpathian Control Conference (ICCC)*, pages 813–818, 2016. doi: 10.1109/CarpathianCC.2016.7501208.

[36] V. Záda and K. Belda. Application of Hamiltonian mechanics to control design for industrial robotic manipulators. In *2017 22nd International Conference on Methods and Models in Automation and Robotics (MMAR)*, pages 390–395, Miedzyzdroje, Poland, 28–31 Aug. 2017. doi: 10.1109/MMAR.2017.8046859.

[37] K. Chadaj, P. Malczyk, and J. Frączek. A parallel recursive hamiltonian algorithm for forward dynamics of serial kinematic chains. *IEEE Transactions on Robotics*, 33(3):647–660, 2017. doi: 10.1109/TRO.2017.2654507.

[38] G. Schreiber, A. Stemmer, and R. Bischoff. The fast research interface for the KUKA lightweight robot. In *Proceedings of the IEEE ICRA 2010 Workshop on ICRA 2010 Workshop on Innovative Robot Control Architectures for Demanding (Research) Applications – How to Modify and Enhance Commercial Controllers*, pages 15–21, May 2010.