

The advanced iterative learning control algorithm for rehabilitation exoskeletons

Mihailo Lazarević¹⁾
Nikola Živković²⁾

In this paper an advanced iterative learning control algorithm for rehabilitation exoskeletons is proposed. A simplified biomechanical model is used as the control object to verify control algorithm feasibility. The control design is proposed as two level controller consisting of inner and outer loop. In the inner loop the feedback linearization is applied to cancel out the model nonlinearities. In the outer loop the advanced iterative learning control algorithm of sgnPDD² type is applied as a feed-forward controller and classical PD controller as a feedback controller. Uncertainties are added in order to examine the controller design robustness. Numerical simulation is carried out.

Key words: biomechanical model, iterative learning control, feedback linearization, uncertainty.

Introduction

EXOSKELETON as a real-time interaction with the wearer's intelligent robot, in recent years, becomes a hot topic mouth class research in the field of robotics. The term 'exoskeleton' was used in biology referring to the chitinous or calcified external skeleton used by numerous animal taxa for structural support and defense against predators [1, 2].

Wearable exoskeleton outside the body, combined with the organic body, plays a role in the protection and support. By wearing an exoskeleton robot, it is possible to expand the wearer's athletic ability, increase muscle endurance, and enable the wearer to complete tasks that one cannot perform under natural conditions as well as help for recovering of lost human motor functions. Based on the above advantages, the exoskeleton robot in military medical care and rehabilitation has broad application prospects [2].

In the military field, the exoskeleton is very attractive because it can effectively improve the individual combat capability. For example, HULC is an exoskeleton robot that can greatly increase the soldier's ability to carry weight, making it easy for soldiers to carry heavy loads of 90 kg.

Now, the exoskeletons are generally regarded as a technology that extends, complements, substitutes or enhances human function and capability or empowers the human limb where it is worn [3]. Unlike other robots, the operator of exoskeletons is a human who needs to make decisions [4] and perform tasks with exoskeletons.

Stroke survivors are faced with some degree of limb impairment, depending on the place in brain, structure and size of caused damage. An increasing number of evidence suggests that brain of a patient who survived stroke has increased capacity for plastic change and thus some motor functions can be fully or partially recovered. Rehabilitative training plays an important role in recovery of lost motor functions [5]. In order to enhance a therapy delivered by

therapists, the use of robotics emerged as an aid in the rehabilitation process [6]. In [7] and [8] it is shown that the robot-aided training, especially in the upper limbs, has positive effects in patient stroke rehabilitation. Also, it has been observed that robot-assisted rehabilitation has potential benefits to patients even several years after the stroke.

Rehabilitative robotics of the upper limbs can be differentiated into two types – rehabilitative robots with one point of contact (end-effector robots) and rehabilitative robots with multiple points of contact (exoskeletons). End-effector robot supports patients arm in one point of contact, usually patients hand or forearm. End-effector robot joints movement does not coincide with the movement of the patients arm. These drawbacks with the end-effector robots influenced the research of exoskeleton rehabilitation robots. Exoskeletons mitigate important flaws of the end-effector robots mentioned above [9]. Rehabilitation robots can be developed to assist rehabilitation in individuals with stroke.

Considering the repetitive nature of the patient's rehabilitation process, Iterative Learning Control (ILC) algorithms emerged as suitable for this application. In the last three decades, (ILC) has been extensively studied. ILC has become one of the most active research and study topic in the field of control theory and its applications [10], [11], [12], [13], [14].

ILC is an intelligent control method for systems that perform tasks repetitively over a finite time interval. This method is similar to how humans learn – performing a certain task and observing the outcome. The next performance of the same task will be adjusted accordingly to the previously observed outcome. ILC algorithm constructs current system input by applying predetermined mathematical law on a previous system output which was stored in the memory [11]. Loosely said ILC is using past experience to improve current system behaviour. Basic ILC principle is illustrated in Fig.1.

¹⁾ University of Belgrade, Faculty of Mechanical Engineering, Department of Mechanics, Kraljice Marije 16, 11020 Belgrade 35, SERBIA

²⁾ Lola institute, Kneza Visaslava 70a, 11030 Belgrade, SERBIA

Correspondence to: Milailo Lazarević; e-mail: mihailo.lazarevic@gmail.com

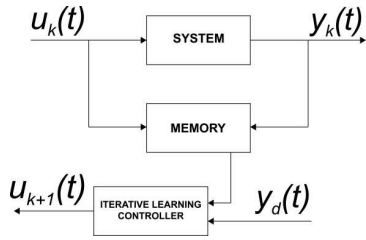


Figure 1. Basic ILC scheme

As it has been already said, rehabilitation training is a kind of repetitive training. Patient's motor function will improve with an increase in the number of training while the auxiliary level of robot and electrical stimulation will be reduced. In ILC the control input is directly updated between trials and it is this feature that makes it suitable for exoskeleton robots (i.e. robotic assisted stroke rehabilitation) [15].

In this paper, an advanced robust open-closed iterative learning control for exoskeleton rehabilitation robots is presented.

Control object model

In order to validate the proposed control algorithm feasibility, the control object is mathematically modelled using the Rodriguez approach. Human upper limb and supporting exoskeleton are adopted as the control object in this paper. The existing complex system is simplified to a biomechanical model consisting of 3DOF robot arm with truncated cone shaped links (Fig.2).

Structure of this biomechanical model is a sequence of two rigid bodies – links, interconnected with joints where the first link has two degrees of freedom and the second link has one degree of freedom. In order to apply the Rodriguez approach the first joint is decomposed from 2DOF to two links interconnected with 1DOF joints. The first link is set to be fictive, i.e. mass and length of this link is zero. Now, open chain system of rigid bodies consists of three bodies with joints between them starting from inertial reference frame $Oxyz$ as shown in Figure . The values $q^i, i=1,2,3$ are generalized coordinates that define a configuration of the mechanical model. Each link is associated with local reference frame $C\xi\eta\zeta$.

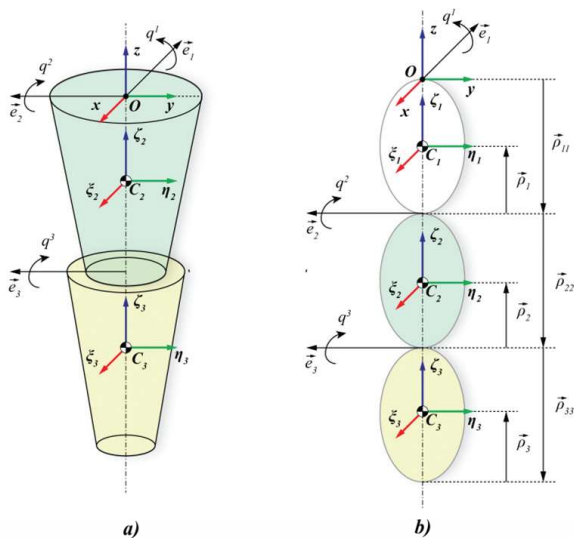


Figure 2. a)-Simplified model of the exoskeleton-human arm, b) kinematic scheme after joint decomposition

At initial time reference configuration is set by setting corresponding axes of reference frames parallel to the Cartesian inertial reference frame $Oxyz$. This convention enables using the Rodriguez approach for obtaining dynamic equations of motion.

Further, parameters $\xi_i, \bar{\xi}_i = 1 - \xi_i$ are defined, which denote whether a joint is prismatic or cylindrical. The geometry of the model is defined by the unit vectors \bar{e}_i and the position vectors \bar{p}_i and \bar{p}_{ii} expressed in local coordinate frames $C_i\xi_i\eta_i\zeta_i$. Origins of these local coordinate frames are attached to the centre of mass of each link i [16], [17].

Dynamic equations of motion for the robot system can be obtained by applying the Lagrange equations of the second kind in the covariant form as follows:

$$\sum_{\alpha=1}^n a_{\gamma\alpha}(q)\ddot{q}_\alpha + \sum_{\alpha=1}^n \sum_{\beta=1}^n \Gamma_{\alpha\beta,\gamma} \dot{q}_\alpha \dot{q}_\beta = Q_\gamma \quad (1)$$

where the coefficients $a_{\gamma\alpha} = \alpha_{\alpha\gamma}$ are the covariant coordinates of the basic metric tensor and present the Christoffel symbols of the first kind. Coefficients of the metric tensor are defined as,

$$\alpha_{\alpha\beta} = \sum_{i=1}^n m_i (\bar{\mathbf{T}}_{a(i)}) \{ \bar{\mathbf{T}}_{\beta(i)} \} + (\bar{\mathbf{\Omega}}_{a(i)}) [J_{Ci}] \{ \bar{\mathbf{\Omega}}_{\beta(i)} \} \quad (2)$$

where quasi-base vectors $\bar{\mathbf{T}}_{a(i)}$ and $\bar{\mathbf{\Omega}}_{a(i)}$ are:

$$\bar{\mathbf{T}}_{a(i)} = \begin{cases} \bar{\xi}_\alpha \bar{e}_\alpha \times \bar{\mathbf{R}}_{a(i)} + \xi_\alpha \bar{e}_\alpha, & \forall \alpha \leq i \\ 0, & \forall \alpha > i \end{cases} \quad (3)$$

$$\bar{\mathbf{\Omega}}_{a(i)} = \begin{cases} \bar{\xi}_\alpha \bar{e}_\alpha, & \forall \alpha \leq i \\ 0, & \forall \alpha > i \end{cases} \quad (4)$$

where $\bar{\mathbf{R}}_{a(i)} = \sum_k^i = \alpha (\bar{p}_{kk} + \xi_k \bar{e}_k q^k) + \bar{p}_i$. The Christoffel symbols are:

$$\Gamma_{\alpha\beta,\gamma} = \frac{1}{2} \left(\frac{\partial a_{\beta\gamma}}{\partial q^\alpha} + \frac{\partial a_{\gamma\alpha}}{\partial q^\beta} - \frac{\partial a_{\alpha\beta}}{\partial q^\gamma} \right), \alpha, \beta, \gamma = 1, \dots, n \quad (5)$$

The generalized forces can be presented as:

$$\mathbf{Q}_\gamma = \mathbf{Q}_\gamma^a + \mathbf{Q}_\gamma^g + \mathbf{Q}_\gamma^v + \mathbf{Q}_\gamma^c + \mathbf{Q}_\gamma^f \quad (6)$$

Wherein $\mathbf{Q}_\gamma^a, \mathbf{Q}_\gamma^g, \mathbf{Q}_\gamma^v, \mathbf{Q}_\gamma^c$ and \mathbf{Q}_γ^f denote the generalized control, gravitational, viscous, spring and friction forces, respectively.

Biomechanical model dynamics can be written in compact matrix form:

$$a(\mathbf{q}) \ddot{\mathbf{q}} + \left(K(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{Q}^g \right) = a(\mathbf{q}) \dot{\mathbf{q}} + \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{Q}^u \quad (7)$$

where $a(\mathbf{q})$ is the inertia matrix, $K(q, \dot{q})$ is matrix that includes centrifugal and the Coriolis effects, \mathbf{Q}^g is vector of gravitational forces and \mathbf{Q}^u is vector of generalized control forces [18]. It is assumed that viscous, spring and friction forces are equal to zero in our case.

Control design

The obtained mathematical model is multi-input multi-output, nonlinear, time-varying system. Control law design in this case consists of inner and outer control loop. Inner loop is Feedback linearizing control law and outer loop is Iterative Learning Control algorithm with classical PD feedback.

Feedback Linearization

Feedback linearization or Computed torque control algorithm is introduced in order to linearize nonlinear biomechanical model so the linear control laws can be applied in the outer loop. This linearization is exact and it differs from the Jacobian linearization which is linear approximation of dynamics of given system. Feedback linearization can be applied in general form [19], but here the concept of Computed torque is applied out of the convenience. Given the equations of motion (7),

$$a(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{Q}^u \quad (8)$$

where input vector \mathbf{Q}^u can be chosen as:

$$\mathbf{Q}^u = a(\mathbf{q})\mathbf{u} + \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) \quad (9)$$

Where \mathbf{u} is new input vector. A new linear decoupled system is obtained in form of double integrator:

$$\ddot{\mathbf{q}} = \mathbf{u} \quad (10)$$

Now \mathbf{u} can be chosen conveniently as the linear control law which is going to close the outer control loop.

Open-closed loop iterative learning control

After closing the inner loop control algorithm in the outer loop an open-closed loop ILC algorithm is applied which consists of feed-forward sgnPDD^2 type control law and feedback PD type control law as represented in the block diagram (Fig.3).

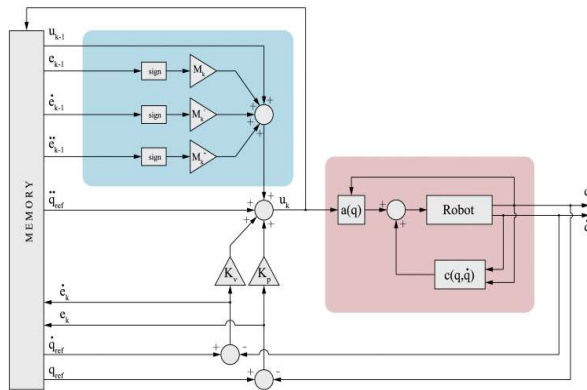


Figure 3. Block diagram of an ILC sgnPDD^2/PD type control algorithm

Control law can be written as:

$$\begin{aligned} \mathbf{u}_k(t) &= \mathbf{u}_{\text{kff}}(t) + \mathbf{u}_{\text{kfb}}(t) \\ &= \mathbf{u}_{k-1} + \text{sgn}(\mathbf{e}_{k-1})M_k + \text{sgn}(\dot{\mathbf{e}}_{k-1})M_k^{\dot{}} \\ &\quad + \text{sgn}(\ddot{\mathbf{e}}_{k-1})M_k^{\ddot{}} + K_p\mathbf{e}_k + K_D\dot{\mathbf{e}}_k \end{aligned} \quad (11)$$

where $\mathbf{e}_k(t) = \mathbf{y}_d(t) - \mathbf{y}_k(t)$ is trajectory tracking error and $\mathbf{y}_d(t)$ is desired output trajectory. K_p and K_D are closed-loop positive definite diagonal learning matrices. Function $\text{sgn}(\cdot)$ is sign function defined as:

$$\text{sgn}(x) = \begin{cases} -1, x < 0 \\ 0, x = 0 \\ 1, x > 0 \end{cases} \quad (12)$$

and value k in index denotes the iteration number. Advantages of the proposed approach in calculating open-loop ILC law are: the controller needs only addition and subtraction as well as the controller needs only to store the values of corresponding variables, thus dramatically reduces the demanding for storage size.

$M_k, M_k^{\dot{}}$ and $M_k^{\ddot{}}$ are the functions of $|se_k|, |s\dot{e}_k|, |s\ddot{e}_k|$, respectively [20]. These sums are defined as:

$$se_k = \sum_0^T \mathbf{e}_k(t), s\dot{e}_k = \sum_0^T \dot{\mathbf{e}}_k(t), s\ddot{e}_k = \sum_0^T \ddot{\mathbf{e}}_k(t) \quad (13)$$

Also, to reduce the computation and storage size for the proposed method, parameters, $M_k, M_k^{\dot{}}$ and $M_k^{\ddot{}}$ can be adapted as a simple step functions of previously defined error sums (13). This is shown in Fig.4.

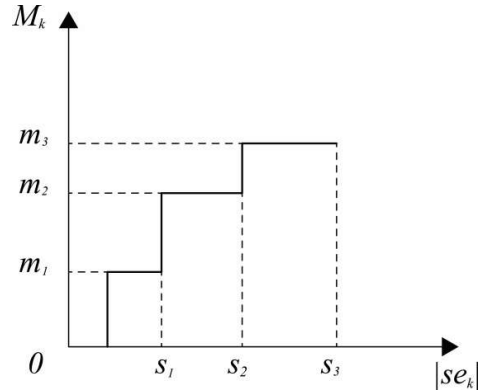


Figure 4. Step function M_k . Threshold values $m_1, m_2, m_3, s_1, s_2, s_3$ initially are chosen arbitrarily and then tuned.

For example if sum $|se_k|$ has value between S_1 and S_2 , parameter M_k is designated with value m_2 . Parameters $M_k^{\dot{}}$ and $M_k^{\ddot{}}$ are chosen analogously. Step function threshold values are chosen purely by trial and error and currently no specific recommendation can be given for the initial selection.

Simulation results and discussion

In this section the numerical simulation in MATLAB is carried out in order to verify control law error convergence. A method used to solve ordinary differential equations is the Runge-Kutta with fixed step. Simplified exoskeleton/human arm model with 3DOF is used as a simulation object with following parameters $m_1 = 0\text{ kg}$, $m_2 = 1.4\text{ kg}$, $m_3 = 1.1\text{ kg}$, $l_1 = 0\text{ m}$, $l_2 = 0.369\text{ m}$, $l_3 = 0.36\text{ m}$ where the first segment is fictive due to 2DOF joint decomposition [16]. Exoskeleton system is tasked with desired trajectory tracking, $\mathbf{q}_d(t) \in \mathbb{R}^n$. System response is observed over the time interval $t = [0, T]$ where $T = 5\text{ s}$. Desired trajectories are given as the fifth order

polynomials $q_d(t) = a_0 + a_1t + a_2t^2 + a_3t^3 + a_4t^4 + a_5t^5$ with constraints:

$$\begin{aligned} q_{dk}(0) &= 0, k = 1, 2, 3; \\ q_{d1}(T) &= \frac{\pi}{2}, q_{d2}(T) = \frac{\pi}{4}, q_{d3}(T) = \frac{\pi}{6}; \\ \dot{q}_{dk}(0) &= \dot{q}_{dk}(T) = 0, k = 1, 2, 3; \\ \ddot{q}_{dk}(0) &= \ddot{q}_{dk}(T) = 0, k = 1, 2, 3; \end{aligned} \quad (14)$$

Let the error bounds be $\max(e) \leq \pm 0.5 \cdot 10^{-2}$ rad. Exact feedback linearization of the given system left us with a linear system. In order to verify the control law robustness additional uncertainties are introduced in the system via $\boldsymbol{\eta}$. Uncertainty of the given biomechanical model can be represented as a change of mass of the object. With that in mind inertia matrix, centrifugal matrix and gravity vector can be written as a sum of nominal and uncertain part [21]. Equations of motion (7) can be rewritten as:

$$(a_N(\mathbf{q}) + \Delta a) \ddot{\mathbf{q}} + (K_N(\mathbf{q}, \dot{\mathbf{q}}) + \Delta K) - (\mathbf{Q}_N^g + \Delta \mathbf{Q}^g) = \mathbf{Q}^u \quad (15)$$

Applying computed torque we can choose:

$$\mathbf{Q}^u = a_N(\mathbf{q})\mathbf{u} + K_N(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{Q}_N^g \quad (16)$$

Equations now become:

$$\ddot{\mathbf{q}}(\mathbf{t}) = a^{-1}(\mathbf{q})a_N(\mathbf{q})\mathbf{u}(\mathbf{t}) + \boldsymbol{\eta}(\mathbf{q}, \dot{\mathbf{q}}) \quad (17)$$

where \mathbf{u} is a new control signal and $\boldsymbol{\eta}$ is the model uncertainty:

$$\boldsymbol{\eta}(\mathbf{q}, \dot{\mathbf{q}}) = -a^{-1}(\mathbf{q})(\Delta \mathbf{K}(\mathbf{q}, \dot{\mathbf{q}}) - \Delta \mathbf{Q}^g) \quad (18)$$

A new control input \mathbf{u} is selected as *sgn**PDD*²/*PD* type controller designed in the previous section. Equations of motion can be represented in state-space form as:

$$\begin{aligned} \dot{\mathbf{x}}(\mathbf{t}) &= \mathbf{A}\mathbf{x}(\mathbf{t}) + \mathbf{B}_a\mathbf{u}(\mathbf{t}) + \mathbf{D}\boldsymbol{\eta}(\mathbf{q}, \dot{\mathbf{q}}), \\ \mathbf{y}(\mathbf{t}) &= \mathbf{C}\mathbf{x}(\mathbf{t}). \end{aligned} \quad (19)$$

Where

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} \mathbf{0}_{n \times n} & \mathbf{I}_{n \times n} \\ \mathbf{0}_{n \times n} & \mathbf{0}_{n \times n} \end{bmatrix}, \mathbf{B}_a = \begin{bmatrix} \mathbf{0}_{n \times n} \\ a^{-1}a_N \end{bmatrix}, \mathbf{D} = \begin{bmatrix} \mathbf{0}_{n \times n} \\ \mathbf{I}_{n \times n} \end{bmatrix}, \\ \mathbf{C} &= [\mathbf{I}_{n \times n} \quad \mathbf{0}_{n \times n}]. \end{aligned} \quad (20)$$

Before applying the proposed ILC algorithm as input \mathbf{u} , the following assumptions will be made:

- (i) Desired trajectories $\mathbf{q}_d(\mathbf{t})$ are continuously differentiable on $[0, T]$,
- (ii) Initial conditions for all iterations are

$$\mathbf{x}_k(0) = \mathbf{x}_d(0), k = 1, 2, \dots, \quad (21)$$

- (iii) Influence of changing the masses of links are negligible on matrix $a(\mathbf{q})$, so it follows that $a^{-1}a_N \approx \mathbf{I}$ and

$$\mathbf{B}_a = \mathbf{B} = \begin{bmatrix} \mathbf{0}_{n \times n} \\ \mathbf{I}_{n \times n} \end{bmatrix},$$

(iv) System (18) is causal [17].

Gain matrices for the feedback loop are chosen as $K_P = \text{diag}\{161010\}$ and $K_D = \text{diag}\{857\}$. Step values M_k , M'_k and M''_k and their bounds are chosen in the process of trial and error:

$$M_{k1} = \begin{cases} 0.9, |se_k| > 20 \\ 0.03, 1.5 < |se_k| \leq 20 \\ 0.001, |se_k| \leq 1.5 \end{cases}$$

$$M_{k2} = \begin{cases} 0.5, |se_k| > 15 \\ 0.05, 2 < |se_k| \leq 15 \\ 0.001, |se_k| \leq 2 \end{cases} \quad (22)$$

$$M_{k3} = \begin{cases} 0.1, |se_k| > 15 \\ 0.05, 1 < |se_k| \leq 15 \\ 0.001, |se_k| \leq 1 \end{cases}$$

$$M'_{k1} = \begin{cases} 0.01, |se_k| > 0.2 \\ 0.005, 0.1 < |se_k| \leq 0.2 \\ 0.001, |se_k| \leq 0.1 \end{cases}$$

$$M'_{k2} = \begin{cases} 0.01, |se_k| > 0.5 \\ 0.005, 0.1 < |se_k| \leq 0.5 \\ 0.001, |se_k| \leq 0.1 \end{cases} \quad (23)$$

$$M'_{k3} = \begin{cases} 0.009, |se_k| > 0.5 \\ 0.005, 0.25 < |se_k| \leq 0.5 \\ 0.001, |se_k| \leq 0.25 \end{cases}$$

$$M''_{k1} = \begin{cases} 0.01, |se_k| > 0.5 \\ 0.005, 0.1 < |se_k| \leq 0.5 \\ 0.001, |se_k| \leq 0.1 \end{cases}$$

$$M''_{k2} = \begin{cases} 0.09, |se_k| > 0.5 \\ 0.005, 0.1 < |se_k| \leq 0.5 \\ 0.001, |se_k| \leq 0.1 \end{cases} \quad (24)$$

$$M''_{k3} = \begin{cases} 0.008, |se_k| > 0.5 \\ 0.005, 0.25 < |se_k| \leq 0.5 \\ 0.001, |se_k| \leq 0.25 \end{cases}$$

It can be seen from figures (Figures 1, 2 and 3) that the system output reaches desired trajectory. Increasing mass uncertainties results in slower system output error convergence.

The first link (fictive) reaches proposed error bound in 35 iterations for 5% mass uncertainty, in 96 iterations for 10% mass uncertainty and in 101 iterations for 15% mass uncertainty. Two other links reach desired trajectory much faster, within 20 iterations.

Also, it can be concluded that, although the system response in some cases requires relatively large number of iterations to reach the desired error bound, the initial trajectory does not deviate too much from the desired trajectory. In other words, maximum errors above the proposed error bound are not drastically greater than mentioned error bound for real world applications, in this case - patient rehabilitation.

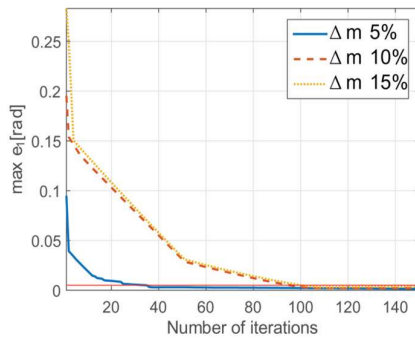


Figure 5. Maximum error of the first link per iteration for different percentages of link mass uncertainties

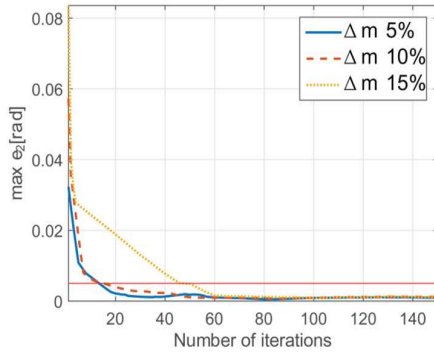


Figure 6. Maximum error of the second link per iteration for different percentages of link mass uncertainties

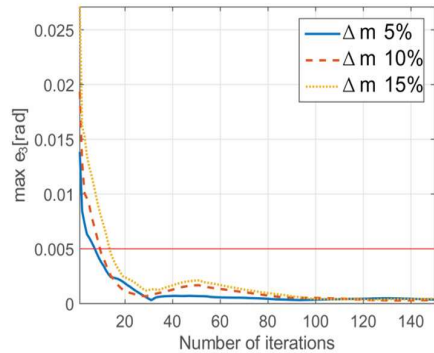


Figure 7. Maximum error of third link per iteration for different percentages of link mass uncertainties

Conclusion

In this paper $\text{sgn}PDD^2$ type of feed-forward iterative learning controller is investigated. In addition to feed-forward controller, regular PD feedback controller is present. Control law behaviour is tested, through simulation, on a linear system with added bounded uncertainties. A simulation results show convergence of the systems response and controller ability to track the reference trajectory. System output in some cases needs a relatively large number of iterations to reach the desired trajectory.

Parameters M_k , M'_k , M''_k and threshold values in $\text{sgn}PDD^2$ type of ILC algorithm are chosen arbitrarily and tuned by trial and error, which can be time consuming.

Acknowledgment

The presented research was supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia by contract no. 451-03-9/2021-14/200105 from 05.02.2021 and contract no. 451-03-9/2021-14/200066.

Literature

- [1] SASHI S KOMMU Ed. Rehabilitation Robotics, I-Tech Education and Publishing, Vienna Austria, 2007.
- [2] GUNASEKARA, J.M.P., GOPURA, R.A.R.C., JAYAWARDANE, T.S.S., LALITHARATHNE, S.T.: *Control Methodologies for Upper Limb Exoskeleton Robots*, 2012 IEEE/SICE International Symposium on System Integration (SII) Kyushu University, Fukuoka, Japan December 16-18, 19-24 (2012).
- [3] YANG, C.J., ZHANG, J.F., CHEN, Y., DONG, Y.M., ZHANG, Y.: *A Review of exoskeleton-type systems and their key technologies*, Proceedings of The Institution of Mechanical Engineers Part C-journal of Mechanical Engineering Science 222, 1599-1612 (2008).
- [4] ZHENG, Y., SONG, Q., LIU, J., SONG, Q., YUE, Q.: *Research on motion pattern recognition of exoskeleton robot based on multimodal machine learning model*, Neural Computing and Applications (2020) 32:1869-1877.
- [5] LOSSEFF, N., THOMPSON, A.J., Eds.: *Neurological Rehabilitation of Stroke*. Taylor & Francis, UK, 2004.
- [6] TEASELL, R.W., LALIT, K.: *What's New in Stroke Rehabilitation*, AHAJ, no.35, pp.383-385, 2004, doi: 10.1161/01.STR.0000115937.94104.76.
- [7] VOLPE, B., KREBS, H., HOGAN, N.: *Robot-aided sensorimotor training in stroke rehabilitation*, Adv. Neurol., vol. 92, pp. 429-33, Feb. 2003.
- [8] KREBS, H., VOLPE, B., HOGAN, N., AISEN, M.: *Increasing productivity and quality of care : Robot-aided neuro-rehabilitation*, Journal of Rehabilitation Research and Development, vol. 37, no. 6, pp. 639-652, Dec. 2000.
- [9] LO, H.S., XIE, S.Q.: *Exoskeleton robots for upper-limb rehabilitation: State of the art and future prospects*, Med. Eng. Phys., vol. 34, no. 3, pp. 261-268, Apr. 2012, doi: 10.1016/j.medengphy. 2011.10.004.
- [10] ARIMOTO, S., KAWAMURA, S., MIYAZAKI, F.: *Bettering operation of Robots by learning*, J. Robot. Syst., vol. 1, no. 2, pp. 123-140, 1984, doi: https://doi.org/10.1002/rob.4620010203.
- [11] AHN, H.S., MOORE, K.L., CHEN, Y.: *Iterative Learning Control: Robustness and Monotonic Convergence for Interval Systems*. London: Springer-Verlag, 2007.
- [12] XU, J.X.S., PANDA, K., LEE, T.H.: *Real-time Iterative Learning Control: Design and Applications*. London: Springer-Verlag, 2009.
- [13] LAZAREVIĆ, M.: *Iterative learning control of integer and noninteger order: An overview*, Sci. Tech. Rev., vol. 64, no. 1, pp. 35-47, 2014.
- [14] LAZAREVIĆ, P.M., TZEKIS, P.: *Robust second-order PDA type iterative learning control for a class of uncertain fractional order singular systems*, J. Vib. Control, vol. 22, no. 8, pp. 2004-2018, May 2016, doi: 10.1177/1077546314562241.
- [15] FREEMAN, C., ROGERS, E., BURRIDGE, J.H., HUGHES, A.M., MEADMORE, K.L.: *Iterative Learning Control for Electrical Stimulation and Stroke Rehabilitation*. London: Springer-Verlag, 2015.
- [16] ČOVIĆ, V., LAZAREVIĆ, M.: *Robot mechanics*. Faculty of Mechanical Engineering, Belgrade, 2007.
- [17] LAZAREVIĆ, M.P.: *Mechanics of Human Locomotor System*, FME Transactions, vol. 34, no. 2, pp. 105-114, 2006.
- [18] SICILIANO, B., SCIAVICCO, L., VILLANI, L., ORIOLO, G.: *Robotics: Modelling, Planning and Control*. London: Springer-Verlag, 2009.
- [19] SLOTINE, J.: *Applied nonlinear control*. Englewood Cliffs N.J.: Prentice Hall, 1991.
- [20] WANG, Y.: *Iterative Learning Control Algorithm with Self-adaptive Steps*, Proceedings of the 10th World Congress on Intelligent Control and Automation, Jul. 2012.
- [21] LAZAREVIĆ, P.M., MANDIĆ, P., OSTOJIĆ, S.: *Further results on advanced robust iterative learning control and modeling of robotic systems*, Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci., p. 0954406220965996, Nov. 2020, doi: 10.1177/0954406220965996.

Napredni iterativni algoritam upravljanja učenjem za rehabilitacione egzoskelete

U ovom radu predložen je napredni iterativni algoritam kontrole učenja za rehabilitacione egzoskelete. Pojednostavljeni biomehanički model koristi se kao objekt upravljanja kako bi se proverila izvodljivost algoritma upravljanja. Projektovano upravljanje je predloženo kao dvostepeni regulator koji se sastoji od unutrašnje i spoljašnje petlje. U unutrašnjoj petlji primjenjuje se linearizacija kako bi se eliminisala nelinearnost modela. U spoljašnjoj petlji napredni iterativni algoritam upravljanja učenja tipa sgnPDD2 primjenjuje se kao kontroler u direktnoj grani, a klasični PD kontroler kao kontroler u povratnoj grani. Dodaju se nesigurnosti kako bi se ispitala robusnost projektovanog kontrolera. Sprovedena je numerička simulacija.

Ključne reči: biomehanički model, iterativna kontrola učenja, feedback linearizacija, neizvesnost.