

# Modeling and Networked Control of Two-rigid link Robot Arm

OULD MOHAMED MOHAMED VALL

Department of Computer Engineering and Networks  
College of Computer and Information Sciences,  
Jouf University,  
Sakaka 75471,  
Kingdom of Saudi Arabia,

*Abstract*-A networked control system (NCS) is one in which controller(s), actuator(s), and sensor(s) exchange command signals and data through a limited-bandwidth communication network that may be used by other applications, devices, and control systems. Compared to classical wired controlled systems, NCSs possess many advantages. In this paper, we propose the modeling and networked control of two-rigid link robot arm. To deal with the time delays that may occur during communication between the components of the system through the network, a model of the system was first determined, and second, PID controllers were designed based on the obtained model and using the stability region boundary locus technique. To demonstrate the validity of the proposed approach, numerical simulations were conducted using TrueTime, Simscape, SimMechanics, and Simulink with the MATLAB environment.

*Key-words* : Networked control system; Induced-network delay; Two-rigid link robot arm; PID controller; TrueTime; Simscape, SimMechanics.

Received: May 28, 2020. Revised: August 17, 2020. Accepted: September 3, 2020. Published: September 15, 2020.

## 1. Introduction

Robots are becoming increasingly present in people's everyday lives. For example, robots are used for domestic tasks as well as in industry to perform routine, repetitive, or dangerous tasks, to lessen manufacturing costs, and to improve product efficiency and quality. While performing tasks, a robot, or a part of it, is commanded by a controller. The last one should be carefully pre-designed to ensure that the robot performs the required tasks with high degrees of precision, accuracy, and reliability. The problems inherent to the design of a good controller for a robot or part of it have been—and still are—the subject of active research (see [1-12] and references therein). A considerable volume of the research in the literature that relates to the control of robots or their parts is focused on the control of robot arms (see [13-20] and references therein).

On the other hand, compared to classical wired controlled systems, NCSs possess several advantages, such as reduced system wiring, low costs, shared resources, and ease of installation and

maintenance [21, 22, 23]. Therefore, NCSs are steadily becoming more prevalent in various fields, such as traffic management systems, heating control systems, automated highway systems, unmanned aerial vehicles, remote surgery, haptics collaboration over the Internet, and mobile sensor networks [24].

Motivated by the above discussions, in this paper, we propose the modeling and networked control of two-rigid link robot arm. Moreover, to the best of our knowledge, there is scant literature dealing with the networked control of robot arms.

In the research proposed in this paper, we considered that (i) the first link is attached to the robot base by a revolute joint actuated by a DC motor, while the second link is attached to the first link by a revolute joint actuated by another DC motor; (ii) two PID controllers are to be designed to allow the end effector of the robot arm to move from an initial position to a desired position; (iii) each link is considered as an independently networked controlled system, but the two NCSs use the same network. To deal with the time delays that

may occur during the data exchange between the components of a link subsystem through the network, we propose a solution that consists of two steps: first, a model is determined; and second, a PID controller is designed based on the obtained model and using the stability region boundary locus technique.

## 2. Review of NCSs Structure

Fig.1. shows the general structure of a single-loop NCS. As can be seen in Fig.1., the feedback loop is closed over a communication network. As a result, two types of network-induced delays are identified based on where they occur:

- (i)  $\tau^{sc}$  is a network-induced delay from the sensor to the controller, that is, a backward channel delay.
- (ii)  $\tau^{ca}$  is a network-induced delay from the controller to the actuator, that is, a forward channel delay.

In most cases, these two network-induced delays are not treated separately, and only the round-trip delay is of interest [25].

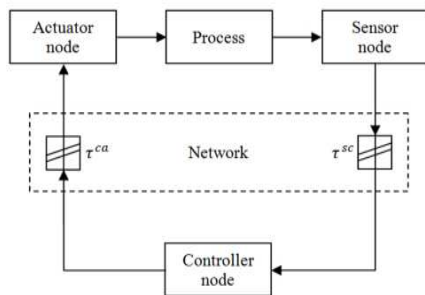


Fig. 1. Single-loop NCS layout

Network-induced delays constitute one of the major disadvantages of NCSs. The delays may degrade the performances of the system or even lead to its instability if they are not taken into consideration during the controller design, and a compensation strategy is not implemented. Many techniques and approaches to compensate for time delays in NCSs have therefore been proposed in the literature(see[26-30] and references therein). In this study, a stability region locus-based technique is proposed to design PID Controllers to compensate for network-induced time delays in NCSs. The proposed technique requires a model for the process to be controlled. Thus, before presenting the

proposed approach to the design of the PID controller, the dynamic model of the two-rigid-link robot arm is reviewed in the section that follows.

## 3. Dynamic Model

Fig.2. shows the schematic diagram of a two-rigid-link robot arm driven by DC motors, where  $L_i$ ,  $m_i$ , and  $\theta_i$   $\{i=1,2\}$  are, respectively, the length, the mass, and the joint angle of the first link  $\{i=1\}$  and of the second link  $\{i=2\}$ ,  $g$  is the gravitational force, and  $V_1$  and  $V_2$  are the voltages that power the DC motors. The DC motors that drive the robot arm deliver torques  $T_1$  and  $T_2$  to the joints, thereby causing the links to rotate by the angles  $\theta_1$  and  $\theta_2$  and move the robot arm end effector from the initial position to the desired position.

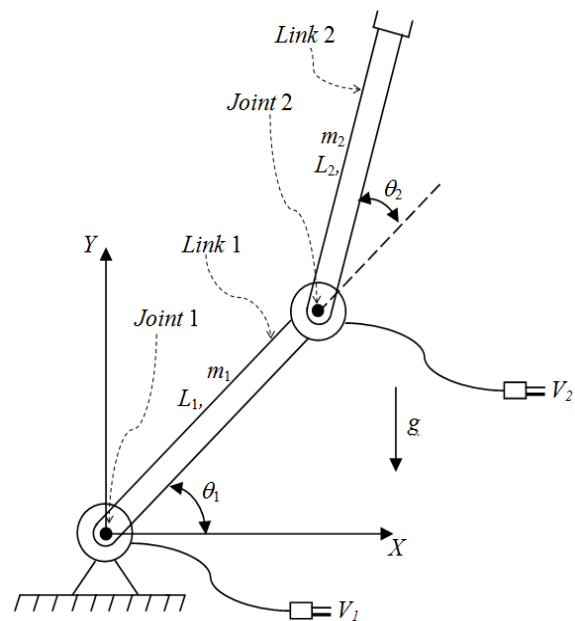
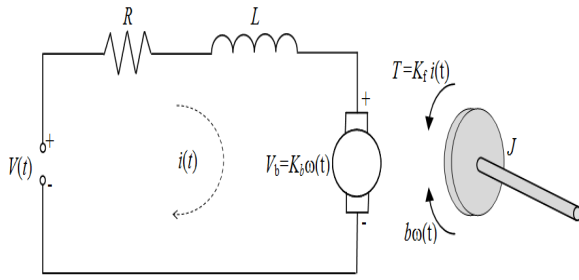


Fig. 2. Schematic diagram of a two-rigid link robot arm driven by DC motors[31]

The schematic representation of a DC motor is shown in Fig. 3, where  $R$  and  $L$  are the armature inductance and resistance, respectively, and  $V(t)$  and  $i(t)$  are the input voltage and the input current, respectively.  $V_b$  and  $T$  are the back electromotive force of the motor and the load torque, respectively, and  $K_b$ ,  $K_f$ ,  $J$ , and  $b$  are the constant of the back electromotive force, the motor constant, the inertia moment, and the friction coefficient of the output shaft, respectively.  $\omega$  is the angular speed of the motor shaft.



**Fig. 3. Schematic diagram of a DC Motor**

From Fig. 3., the dynamic of the DC motor can be described by the following equations obtained based on Kirchoff's second law combined with Newton's second law:

$$L \frac{di(t)}{dt} + Ri(t) = V(t) - K \omega(t) \quad (1)$$

$$J \frac{d\omega(t)}{dt} + b\omega(t) = K_f i(t) \quad (2)$$

Applying the Laplace transform on equations (1) and (2) results in the following equations:

$$LsI(s) + RI(s) = V(s) - K \omega(s) \quad (3)$$

$$Js\omega(s) + b\omega(s) = K_f I(s) \quad (4)$$

where  $s$  denotes the Laplace operator.

From (4),  $I(s)$  can be written as:

$$I(s) = \frac{(Js + b)\omega(s)}{K_f} \quad (5)$$

By inserting (5) into (3), we can obtain the transfer function from the input voltage  $V(s)$  to the angular velocity  $\omega(s)$  as follows:

$$\frac{\omega(s)}{V(s)} = \frac{K_f}{(Ls + R)(Js + b) + KK_f} \quad (6)$$

Thus, the transfer function from  $V(s)$  to the angular position  $\theta(s)$  can be written as

$$\frac{\theta(s)}{V(s)} = \frac{\frac{K_f}{LJ}}{s(s^2 + \frac{(Lb + RJ)}{LJ}s + \frac{Rb + KK_f}{LJ})} \quad (7)$$

## 4. PID Controller Design

This section addresses the design of a PID controller to compensate for the time delays in NCSs.

Let's consider that the overall transfer function of the closed loop NCS is given by

$$T(s) = \frac{(K_p + \frac{K_i}{s} + K_d s)G_p(s)e^{-\tau s}}{1 + (K_p + \frac{K_i}{s} + K_d s)G_p(s)e^{-\tau s}} \quad (8)$$

where  $s$  denotes the Laplace operator.  $K_p$ ,  $K_i$ , and  $K_d$  are the parameters of the PID controller to be determined,  $G_p(s)$  is the transfer function of the plant to be controlled, and  $\tau$  is the time delay "note that the time delay characteristic of an NCS could be constant, bounded, or even random depending on the network protocols adopted and the chosen hardware" [32]. Here, we assume that the time delay  $\tau$  is constant and known.

Let

$$G_p(s) = \frac{B(s)}{A(s)} \quad (9)$$

where  $B(s)$  and  $A(s)$  are two polynomials in  $s$  with real coefficients.

Inserting  $s = j\omega$  and (9) into (8) and decomposing  $B(s)$  and  $A(s)$  into their even and odd parts lead to

$$T(j\omega) = \frac{\left( K_p - \frac{jK_i}{\omega} + jK_d\omega \right) (B_e + jB_o)\omega e^{-j\tau\omega}}{\left( A_e + jA_o\omega \right) \left[ 1 + \frac{\left( K_p - \frac{jK_i}{\omega} + jK_d\omega \right) (B_e + jB_o)\omega e^{-j\tau\omega}}{(A_e + jA_o\omega)} \right]} \quad (10)$$

where  $B_e$  and  $B_o$  are the even and odd parts of  $B(s)$ , respectively, and  $A_e$  and  $A_o$  are the even and odd parts of  $A(s)$ , respectively.

Using Euler's identity  $e^{-j\tau\omega} = \cos(\tau\omega) - j \sin(\tau\omega)$  in equation (10), we can write the characteristic polynomial of  $T(s)$  as

$$P_c(j\omega) = R_c(\omega) + jI_c(\omega) \quad (11)$$

where

$$\begin{aligned} R_c(\omega) = & -B_o K_d \cos(\tau\omega)\omega^3 + B_e K_d \sin(\tau\omega)\omega^2 \\ & + B_o K_i \cos(\tau\omega)\omega + B_o K_p \sin(\tau\omega)\omega^2 \\ & - B_e K_i \sin(\tau\omega) + B_e K_p \cos(\tau\omega)\omega \\ & + A_e \omega \end{aligned}$$

$$\begin{aligned} I_c(\omega) = & B_o K_d \sin(\tau\omega)\omega^3 + B_e K_d \cos(\tau\omega)\omega^2 \\ & - B_o K_i \sin(\tau\omega)\omega + B_o K_p \cos(\tau\omega)\omega^2 \\ & + A_o \omega^2 - B_e K_i \cos(\tau\omega) - B_e K_p \sin(\tau\omega)\omega \end{aligned}$$

Furthermore, the characteristic equation of  $T(s)$  is

$$R_c(\omega) + jI_c(\omega) = 0 \quad (12)$$

Next, dropping  $\omega$  for simplicity and equating the real and imaginary parts  $R_c(\omega)$  and  $I_c(\omega)$  with 0 result in the following equations system:

$$\begin{cases} M_{11}K_p + M_{12}K_i = N \\ M_{21}K_p + M_{22}K_i = Z \end{cases} \quad (13)$$

where

$$M_{11} = B_o \sin(\tau\omega)\omega^2 + B_e K_p \cos(\tau\omega)\omega,$$

$$M_{12} = B_o \cos(\tau\omega)\omega - B_e \sin(\tau\omega),$$

$$N = (B_o \cos(\tau\omega)\omega^3 + B_e \sin(\tau\omega)\omega^2)K_d - A_e \omega$$

$$M_{21} = B_o \cos(\tau\omega)\omega^2 - B_e \sin(\tau\omega)\omega,$$

$$M_{22} = -(B_o \sin(\tau\omega)\omega + B_e \cos(\tau\omega)),$$

$$Z = -(B_o \sin(\tau\omega)\omega^3 - B_e \cos(\tau\omega)\omega^2)K_d - A_o \omega^2$$

By assigning a fixed value to  $K_d$  and solving the equations system (13) for  $K_p$  and  $K_i$ , we get

$$K_p = \frac{A_e B_o \cos(\tau\omega)\omega^2 + A_e B_e \sin(\tau\omega)\omega - A_e B_e \sin(\tau\omega)\omega + A_e B_e \cos(\tau\omega)}{B_o^2 \omega^2 + B_e^2} \quad (14)$$

$$K_i = \frac{\omega(B_o^2 \omega^2 + B_e^2)K_d + A_e B_o \sin(\tau\omega)\omega^2 - A_e B_o \cos(\tau\omega)\omega + A_e B_e \cos(\tau\omega)\omega + A_e B_e \sin(\tau\omega)}{B_o^2 \omega^2 + B_e^2} \quad (15)$$

Plotting the dependency between the values of  $K_p$  and  $K_i$  given by (14) and (15), respectively, into the  $(K_p, K_i)$  plan defines the stability boundary locus. The obtained curve together with the line  $K_i = 0$  split the  $(K_p, K_i)$  plan into the stable and unstable regions. The decision whether the respective region

represents a stabilizing or unstabilizing area can be determined using a test point within each region [33].

## 5. Numerical Simulations

In this section, numerical simulations were conducted in a MATLAB environment in order to study the modeling and networked control of two-rigid link robot arm and to show the validity of the proposed approach for network-induced time delay compensation.

**First**, a two-rigid link robot arm was implemented using SimMechanics and Simulink/MATLAB tools, as shown in Fig. 4. Since in some cases, a robot's arm specifications and parameters may be unknown or known with uncertainties, we assumed that the parameters of the system are unknown. The implemented system was then run to obtain two series of input voltage and angular velocity  $(V, \omega)$  pairs (i.e., one series per link). The obtained data was then used to determine the models for the links taking into account that the transfer function from the input voltage to the angular velocity is of the second order, as shown by (6) and using the MATLAB identification toolbox.

The identification results were as follows:

$$P_1(s) = \frac{\omega_1(s)}{V_1(s)} = \frac{19.984}{-10^{-6}s^2 - 0.0020s + 1} \quad (16)$$

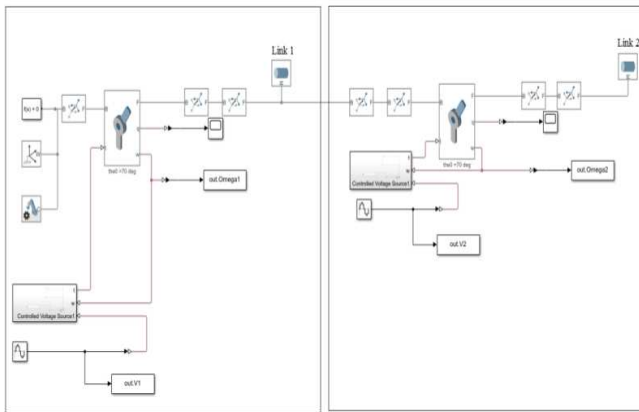
$$P_2(s) = \frac{\omega_2(s)}{V_2(s)} = \frac{20.005}{-1.3678 \cdot 10^{-6}s^2 - 0.0024s + 1} \quad (17)$$

where  $P_1(s)$  is the transfer function from the input voltage  $V_1(s)$  to the angular velocity  $\omega_1(s)$  of the first link, and  $P_2(s)$  is the transfer function from the input voltage  $V_2(s)$  to the angular velocity  $\omega_2(s)$  of the second link.

Then, the transfer function from the input voltage  $V_1(s)$  to the angular position  $\theta_1(s)$  of the first link and the transfer function from the input voltage  $V_2(s)$  to the angular position  $\theta_2(s)$  of the second link were

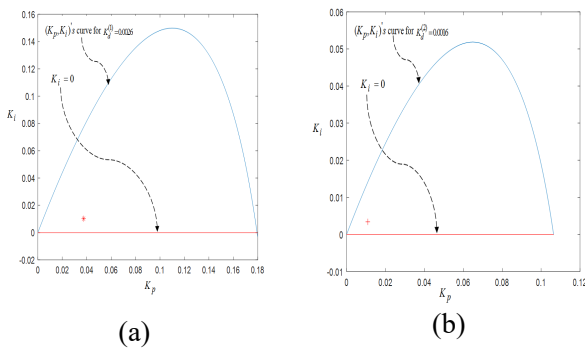
$$G_1(s) = \frac{\theta_1(s)}{V_1(s)} = \frac{19.984}{s(-10^{-6}s^2 - 0.0020s + 1)} \quad (18)$$

$$G_2(s) = \frac{\theta_2(s)}{V_2(s)} = \frac{20.005}{s(-1.3678.10^{-6}s^2 - 0.0024s + 1)} \quad (19)$$



**Fig. 4. Implementation of the two-rigid link robot arm in SimMechanics and Simulink/MATLAB**

**Second**, using the transfer functions(18) and (19), the parameters of the PID controllers were determined based on the approach presented in section 3. Fig. 5. shows the stability regions.



**Fig. 5. Stability regions : (a) stability region for system (18); (b) stability region for system (19)**

The selected values of the PID controllers' parameters were

$$\begin{cases} K_p^{(1)} = 0.0375 \\ K_i^{(1)} = 0.0102 \\ K_d^{(1)} = 0.0102 \end{cases} \quad (20)$$

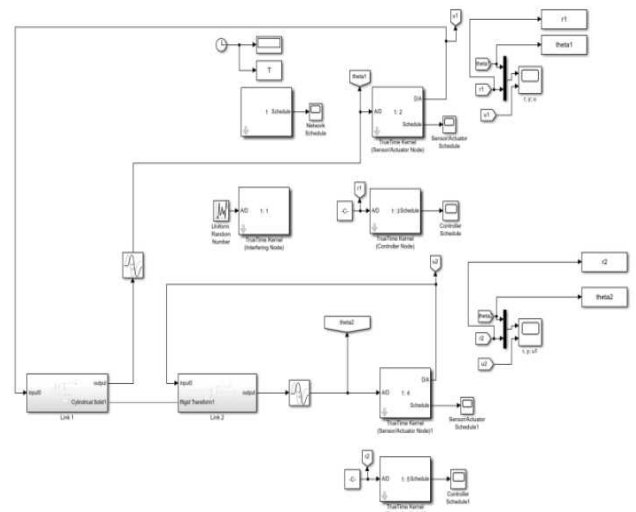
$$\begin{cases} K_p^{(2)} = 0.0109 \\ K_i^{(2)} = 0.0034 \\ K_d^{(2)} = 0.0016 \end{cases} \quad (21)$$

where  $K_p^{(1)}$ ,  $K_i^{(1)}$ , and  $K_d^{(1)}$  are the parameters of the PID controller that controls the first link, and  $K_p^{(2)}$ ,  $K_i^{(2)}$ , and  $K_d^{(2)}$  are the parameters of the PID controller that controls the second link.

**Third**, the model in Fig.4. was included in a TrueTime's Simulink model, as shown in Fig.6. Then, the system was run to obtain the system outputs and to compare them to the desired inputs. During the simulation, the parameters of the simulations were set as follows: the sampling period was 0.01s, the network communication mode was CSMA/CD (Ethernet), the transmission rate was 80Kbit/s, the induced-network time delays were  $\tau^{(1)} = 0.45s$  and  $\tau^{(2)} = 0.75s$ , and the parameters of the controllers in node 2 and node 4 were set as in (20) and (21), respectively. The initial and the desired positions of the first and second links of the robot arm were

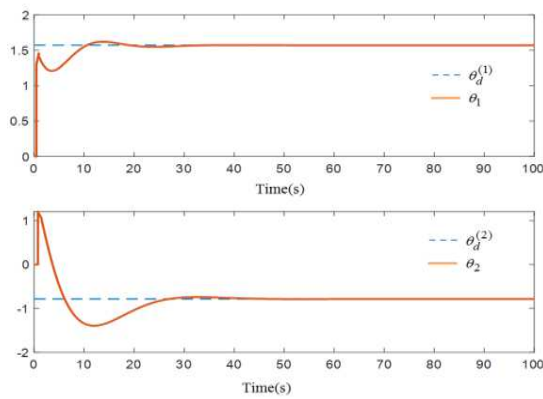
$$\theta_0 = [\theta_1(0) \ \theta_2(0)]^T = \begin{bmatrix} 7\pi/18 & 7\pi/18 \end{bmatrix}^T \text{ and } \theta_d = [\theta_d^{(1)} \ \theta_d^{(2)}]^T = \begin{bmatrix} \pi/2 & \pi/4 \end{bmatrix}^T$$

,respectively.

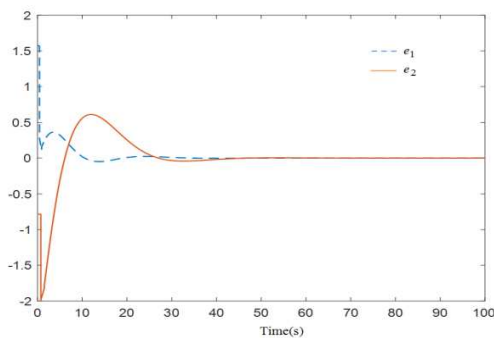


**Fig. 6. Implementation of the networked controlled two-rigid link robot arm in TrueTime, SimMechanics, and Simulink/MATLAB**

Fig. 7. shows the desired positions and the actual positions of the joints 1 and 2. The errors between the desired and actual positions are shown in Fig. 8



**Fig. 7.** The actual and desired positions of the joints, where  $\theta_1$  and  $\theta_d^{(1)}$  are the actual and desired positions of the first joint, respectively, and  $\theta_2$  and  $\theta_d^{(2)}$  are the actual and desired positions of the second joint, respectively.



**Fig.8.** Errors in the angles of the joints, where  $e_1$  is the error in the first joint angle, and  $e_2$  is the error in the second joint angle.

The simulation results in Fig. 7. show the convergence of the actual positions of both joints to their desired values. Fig. 8. shows that the errors in both joints' angles converge to 0. The simulation results therefore demonstrate that the designed PID controllers effectively and efficiently compensated for the network-induced time delays  $\tau^{(1)}$  and  $\tau^{(2)}$  and guaranteed zero steady-state tracking errors of the actual positions  $\theta_1$  and  $\theta_2$  to their respectively desired values  $\theta_d^{(1)}$  and  $\theta_d^{(2)}$ .

## 6. Conclusion

In this paper, we investigated the modeling and networked control of two-rigid link robot arm. The

dynamic modeling of a two-rigid link robot arm driven by DC motors was reviewed. A stability region locus-based approach was proposed to design PID controllers to compensate for the network-induced time delays. Numerical simulations were conducted in MATLAB in order to show the validity of the proposed method. The simulation results demonstrated that the designed controllers could effectively and efficiently compensate for the network-induced time delays and guarantee a very good tracking of the actual outputs of the system to the desired values.

Finally, we notice that it is well established in the literature that networked control systems suffer from two main major problems, namely, network-induced time delays and/or packet loss. In this paper, only the effect of network-induced time delays in NCSs was considered. Therefore, future works will focus on packet loss effect.

Others future works will be devoted to extending the proposed approach to improve the performance of the PID controllers, especially in terms of reducing the time of the convergence of the system's actual outputs to the desired values.

## References

- [1]. Jiang, Y., Yang, C., Wang, Y, Ju, Z.,Li, Y.,& Su, C.-Y. (2020). Multi-hierarchy interaction control of a redundant robot using impedance learning. *Mechatronics*. 67. 102348. doi:10.1016/j.mechatronics.2020.102348
- [2]. Zhang, D.,Kong, L., Zhang, S., Li, Q.,& Fu, Q. (2020). Neural networks-based fixed-time control for a robot with uncertainties and input deadzone. *Neurocomputing*. 390. doi:10.1016/j.neucom.2020.01.072\_
- [3]. Kamedula, M., Kashiri, N.,& Tsagarakis, N. (2020). Wheeled motion kinematics and control of a hybrid mobility CENTAURO robot. *Robotics and Autonomous Systems*. 128. 103482. doi:10.1016/j.robot.2020.103482
- [4]. Kolathaya, S. (2020). Local stability of PD controlled bipedal walking robots. *Automatica*. 114. 108841. doi:10.1016/j.automatica.2020.108841
- [5]. Liu, H., Tao, J., Lyu, P., & Tian, F. (2020). Human-robot cooperative control based on

- sEMG for the upper limb exoskeleton robot. *Robotics and Autonomous Systems*. 125. 103350.  
doi:10.1016/j.robot.2019.103350
- [6]. Chen, D., & Li, S. (2019). A recurrent neural network applied to optimal motion control of mobile robots with physical constraints. *Applied Soft Computing*. 85. 105880. doi: 10.1016/j.asoc.2019.105880
- [7]. Bai, J., Song, A., Wang, T., & Li, H. (2019). A novel back stepping adaptive impedance control for an upper limb rehabilitation robot. *Computers and Electrical Engineering*. 80.  
doi:10.1016/j.compeleceng.2019.106465
- [8]. Ferrara, A., Incremona, G. P., & Sangiovanni, B. (2019). Tracking control via switched integral sliding mode with application to robot manipulators. *Control Engineering Practice*. 90.  
doi:257-266.10.1016/j.conengprac.2019.07.008.
- [9]. Feliu-Talegon, D., Feliu, V., Tejado, I., Vinagre, B., & Hosseinnia, H. (2019). Stable force control and contact transition of a single link flexible robot using a fractional-order controller. *ISA Transactions*.  
doi:10.1016/j.isatra.2018.12.031.
- [10]. Zhu, W., & Sun, Z. Data-based direct fuzzy control for a flexible-link manipulator. 2011 2nd International Conference on Intelligent Control and Information Processing, Harbin, 2011, pp. 87-91, doi:10.1109/ICICIP.2011.6008205.
- [11]. Piotrowski, R. & Maciąg, B. & Makohoń, W. & Milewski, Kr. (2020). Design of Control Algorithms for Mobile Robots in an Environment with Static and Dynamic Obstacles. *Journal of Automation, Mobile Robotics and Intelligent Systems*. 13. 22-30.  
doi:10.14313/JAMRIS/4-2019/34.
- [12]. Jiehua, Z., Yongguo, Z., Cihui Y., (2019). Path Planning and Trajectory Tracking Control of Large Intelligent Mowing Robot based on GPS-RTK, *International Journal of Circuits, Systems and Signal processing*, pp. 132-139, Volume 13.
- [13]. Sharma, K., Jain, N., & Pal, P. (2019). Detection of eye closing/opening from EOG and its application in robotic arm control. *Biocybernetics and Biomedical Engineering*. 40.  
doi:10.1016/j.bbe.2019.10.004.
- [14]. Hou, X., & Tsui, S.-K. (2004). Analysis and control of a two-link and three-joint elastic robot arm. *Applied Mathematics and Computation*. 152. 759-777.  
doi:10.1016/S0096-3003(03)00593-9.
- [15]. Rouzbeh, B., Bone, G., Ashby, G., & Li, E. (2019). Design, implementation and control of an improved hybrid pneumatic-electric actuator for robot arms. *IEEE Access*. 1-1.  
doi:10.1109/ACCESS.2019.2891532.
- [16]. Wei, H., Bu, Y., & Zhu, Z. (2020). Robotic arm controlling based on a spiking neural circuit and synaptic plasticity. *Biomedical Signal Processing and Control*. 55. 101640.  
doi:10.1016/j.bspc.2019.101640.
- [17]. Nazari, M., Rafiee, G., Jafari, A., Golpayegani, H., Reza, S. M. (2008). Supervisory chaos control of a two-link rigid robot arm using OGY method.  
doi: 10.1109/ICCIS.2008.4670980.
- [18]. Marino, A., (2018), Distributed Adaptive Control of Networked Cooperative Mobile Manipulators, *IEEE Transactions on Control Systems Technology*, vol. 26, no. 5, pp. 1646-1660.  
doi:10.1109/TCST.2017.2720673
- [19]. Delavar, R., Tavassoli, B. & Beheshti, M. T. H., (2017), "Improved stability analysis of nonlinear networked control systems over multiple communication links", 2017 Iranian Conference on Electrical Engineering (ICEE), Tehran, 2017, pp. 917-922,  
doi: 10.1109/IranianCEE.2017.7985170.
- [20]. Tavassoli, B., (2011), A continuous-time approach to networked control of nonlinear

- systems, 2011 50th IEEE Conference on Decision and Control and European Control Conference, Orlando, FL, 2011, pp. 217-221, doi: 10.1109/CDC.2011.6161331.
- [21]. Wu, C., Liu, J., Jing, X., Li, H., & Wu, L. (2017). Adaptive fuzzy control for nonlinear networked control systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. 1-11. doi: 10.1109/TSMC.2017.2678760.
- [22]. Baigzadehnoe, B., Rezaie, B., & Rahmani, Z. (2019). Fuzzy-model-based fault detection for nonlinear networked control systems with periodic access constraints and Bernoulli packet dropouts. *Applied Soft Computing*. 80. doi: 10.1016/j.asoc.2019.04.023.
- [23]. Feng, J., Wang, S., & Zhao, Q. (2013). Closed-loop design of fault detection for networked non-linear systems with mixed delays and packet losses. *IET Control Theory & Applications*. 7(6). 858-868. doi:10.1049/iet-cta.2012.0987
- [24]. Mohamed Vall, O.M. (2020). PI controller design for networked control systems with random time delay. *International Journal of Emerging Trends in Engineering Research*. 8. 114-118. doi:10.30534/ijeter/2020/15812020.
- [25]. Zhao, Y.-B., Sun, X.-M., Zhang, J., & Shi, P. (2015). Networked control systems: The communication basics and control methodologies. *Mathematical Problems in Engineering*. 2015. doi:10.1155/2015/639793.
- [26]. Mohamed Vall, O.M. (2020). Artificial neural network-based Smith predictor for compensating random time delays acting in networked control systems. *International Journal of Control and Automation*. 13. 36-44.
- [27]. He, S., Lu, Y., Wu, Y., & Li, Y. (2019). Partial-information-based consensus of network systems with time-varying delay via sampled-data control. *Signal Processing*. 162. doi:10.1016/j.sigpro.2019.04.012.
- [28]. Wang, Yao-Wei & Liu, Andong & Zhang, Wen-An & Yu, Li. (2018). GESO-based control for networked systems with time-varying delays. *Measurement*. 133. doi:10.1016/j.measurement.2018.10.022 .
- [29]. Jin, C. & Liu, B. (2018). Research of sliding mode controller for networked systems with time-varying delay. *Gaojishu Tongxin/Chinese High Technology Letters*. 28. 964-971. doi: 10.3772/j.issn.1002-0470.2018.11-12.010.
- [30]. Min, S., Ling, H., Shen, W., Cui, M., & Xie, W. (2018). Quantized control of event-triggered networked systems with time-varying delays. *Journal of the Franklin Institute*. 356. doi: 10.1016/j.jfranklin.2018.05.041.
- [31]. Reddy, B. R., Reddy, A. C. Dynamic modeling of two-link robot arm driven by DC motors using linear graph theory and principles of mechanics. 2003 National Conference on Trends in Mechanical Engineering, Warangal, 30th August 2003, pp.130-132.
- [32]. Huo, Z. ,& Zhang, Z. (2019). Scheduling and control co-design for Networked Wind Energy Conversion Systems. *Global Energy Interconnection*. 2. 328-335. doi: 10.1016/j.gloi.2019.11.005.
- [33]. Matusšů, R. (2011). Calculation of all stabilizing PI and PID controllers. *International Journal of Mathematics and Computers in Simulation*. 5. 224-231.

### **Creative Commons Attribution License 4.0(Attribution 4.0 International ,CC BY 4.0)**

This article is published under the terms of the Creative Commons Attribution License 4.0 [https://creativecommons.org/licenses/by/4.0/deed.en\\_US](https://creativecommons.org/licenses/by/4.0/deed.en_US)