Analysis and development of a parallelogram linkage manipulator (Phân tích và cải tiến tay máy khớp hình bình hành)

Nguyen Quoc Thinh[1,2], Ngo Ha Quang Thinh[1]
[1]Faculty of Mechanical Engineering, Ho Chi Minh City University of Technology (HCMUT), 268 Ly Thuong Kiet Street, District 10, Ho Chi Minh City, Vietnam
[2]Office for International Study Programs, Ho Chi Minh City University of Technology (HCMUT), 268 Ly Thuong Kiet Street, District 10, Ho Chi Minh City, Vietnam
*Corresponding author: nhqthinh@hcmut.edu.vn

Abstract

In this paper, a control system integrating feedforward, feedback, and iterative learning control (ILC) law is designed for a 3-DOF parallelogram linkage robot. Firstly, the dynamic model is obtained using Newton – Euler formulation, which is then used to calculate the feedforward control signal. The feedback control is designed independently of the feedback control. Subsequently, a passivity-based controller, which is a combination of the two aforementioned, is proposed and compared to the feedforward and feedback alone. Finally, an ILC algorithm is proposed and adjusted according to simulation result to decrease the remaining tracking errors. The performance of four controllers is analyzed and compared in simulation environment based on their tracking process of a reference trajectory.

Bài báo này trình bày kết quả của bộ điều khiển kết hợp yếu tố feedforward, feedback, và iterative learning control (ILC) trên robot khớp hình bình hành 3 bậc tự do. Bài toán động lực học được phân tích bằng phương pháp Newton – Euler và dùng để tính toán tín hiệu điều khiển feedforward cấp vào động cơ. Bộ điều khiển feedback được thiết kế độc lập với bộ feedforward. Sau đó, một bộ điều khiển kết hợp cả hai yếu tố được khảo sát và so sánh với từng bộ điều khiển độc lập. Cuối cùng, thuật toán ILC được thiết kế dựa trên mô phỏng và áp dụng vào tay robot để giảm thiểu sai số bám quỹ đạo. Cả bốn bộ điều khiển được mô phỏng bám một quỹ đạo nhất định và kết quả bám quỹ đạo sẽ được khảo sát, đánh giá và so sánh.

Keywords: *Robot, Parallelogram linkage, Motion control, MATLAB, Iterative learning control, Control system, Simulation.*

1. Introduction

Palletizing robots are currently interested in and developed by manufacturers such as Fuji, ABB, Puma, TMI, etc. Thanks to the use of parallel mechanisms to conduct the end-effector, the weight of the manipulator will be mechanically balanced, allowing for less energy consumption and reduced pressure on the joints and bearings. Currently, palletizing machines are widely used in automatic loading and unloading of crates and packages from the factory conveyor exit to pallets.

Many industrial control applications, especially in the field of robotics, use linear PID controllers for simple and repetitive tasks [1-4]. In some situations, it is difficult to achieve high tracking accuracy using a PID controller. However, with the ability to learn, the controller can compensate for the non-linear elements that classical control techniques are limited in. The solution is to combine the ability to learn from repetitive tasks into the PID controller, allowing the controller to learn from the error of the previous operation to improve the accuracy of the next operation. This technique is called Iterative learning control (ILC) [5-6].

In this paper, the 3-DOF parallelogram palletizing robot is taken as the object of study. The dynamic model is obtained using Newton – Euler formulation in Section 2, which is then used to calculate the feedforward control signal through the actuator modelling in Section 3. The feedback, feedforward, and combination of the two plus ILC are proposed in Section 4. Section 5 analyzes the simulation result on trajectory tracking performance of the four controllers.

2. Dynamic modelling

The main structure of the parallelogram linkage robot is a closed parallel four-bar linkage, it also couples two groups of little parallel four-bar linkages to hold the wrist in term of orientation.

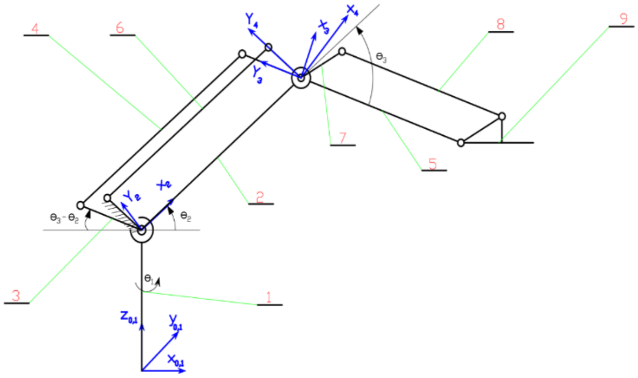

Figure 1 Diagram for dynamics analysis

The dynamic model of the robot manipulator can be described by Lagrange's equation [2]:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau \tag{2.1}$$

Where q is the vector of joint variables, $M(q)$ is the inertia matrix, $C(q,\dot{q})$ is the vector of Coriolis and centrifugal forces, $G(q)$ is the vector of gravity force, τ is the vector of joint control inputs to be designed.

Newton-Euler formulation is used in this paper to solve inverse dynamics problem since it allows computationally efficient computer implementation without the need for differentiation.

For i = 0 to i = 8 do:

$$^{i+1}\omega_{i+1} = {}^{i+1}_{i}R\ {}^{i}\omega_i + \dot{\theta}_{i+1}\, {}^{i+1}\hat{Z}_{i+1} \quad (2.2)$$

$$^{i+1}\dot{\omega}_{i+1} = {}^{i+1}_{i}R\ {}^{i}\dot{\omega}_i + {}^{i+1}_{i}R\ {}^{i}\omega_i\dot{\theta}_{i+1}\, {}^{i+1}\hat{Z}_{i+1} + \ddot{\theta}_{i+1}\, {}^{i+1}\hat{Z}_{i+1} \quad (2.3)$$

$$^{i+1}\dot{v}_{i+1} = {}^{i+1}_{i}R(\ {}^{i}\dot{v}_i + {}^{i}\omega_i \times (\ {}^{i}\omega_i \times {}^{i}P_{i+1}) + {}^{i}\dot{\omega}_i \times {}^{i}P_{i+1}) \quad (2.4)$$

$$^{i+1}\dot{v}_{Ci+1} = {}^{i+1}\dot{v}_{i+1} + {}^{i+1}\omega_{i+1}({}^{i+1}\omega_{i+1} \times {}^{i+1}P_{Ci+1}) + {}^{i+1}\dot{\omega}_{i+1} \times {}^{i+1}P_{Ci+1} \quad (2.5)$$

$$^{i+1}F_{i+1} = m_{i+1}\, {}^{i+1}\dot{v}_{Ci+1} \quad (2.6)$$

$$^{i+1}N_{i+1} = {}^{Ci+1}I\, {}^{i+1}\dot{\omega}_{i+1} + {}^{i+1}\omega_{i+1} \times {}^{Ci+1}I\, {}^{i+1}\omega_{i+1} \quad (2.7)$$

where $^{i+1}\omega_{i+1}$ is angular velocity of link i+1 expressed in {i+1}, $^{i+1}\dot{\omega}_{i+1}$ is acceleration of link i+1 expressed in {i+1}, ^{Ci+1}I is moment of inertia of link i+1, m_{i+1} is mass of link i+1, $^{i+1}_{i}R$ is rotation matrix from {i} to {i+1}, $^{i+1}F_{i+1}$ is inertial force on link i+1 expressed in {i+1}, $^{i+1}N_{i+1}$ is inertial torque on link i+1 expressed in {i+1}, $^{i}P_{i+1}$ is joint position expressed in {i}, $^{i+1}\hat{Z}_{i+1} = [0\ 0\ 1]^T$ is the z-component vector of link i+1.

For i = 9 to i = 1 do:

$$^{i}f_i = {}^{i}F_i - {}_{i+1}^{i}R m_i g + {}_{i+1}^{i}R\, {}^{i+1}f_{i+1} \quad (2.8)$$

$$^{i}n_i = {}^{i}N_i + {}_{i+1}^{i}R\, {}^{i+1}n_{i+1} + {}^{i}P_{i+1} \times {}_{i+1}^{i}R\, {}^{i+1}f_{i+1} + P_{Ci} \times (\ {}^{i}F_i - {}_{i+1}^{i}R m_i g) \quad (2.9)$$

$$\tau_i = {}^{i}n_i^{\ T}\ {}^{i}\hat{Z}_i \quad (2.10)$$

where $^{i}F_i$ is inertial force on link i expressed in {i}, $^{i}N_i$ is inertial torque on link i expressed in {i}, τ_i is the desired actuator torque to rotate link i.

3. Actuator modelling

A DC motor can be controlled by varying the input voltage, commonly by means of changing the PWM duty cycle. In this thesis, to control the output torque, mathematical model of each DC motor should be determined.

The torque τ created by a DC motor is governed by the equation [2]:

$$\tau = k_t I \quad (3.1)$$

where I is the current through the windings, k_t is the torque constant. The back-emf:

$$e = k_e\omega \quad (3.2)$$

where ω is the motor's rotational speed, $k_e = k_t$ is the electrical constant.

According to Kirchhoff's law and Newton's law:

$$L\frac{di}{dt} + Ri = v - k_t\omega \quad (3.3)$$

$$J\frac{d^2\theta}{dt^2} + b\frac{d\theta}{dt} + \tau_{fric} + \tau_D = k_t i \quad (3.4)$$

where θ is the motor's shaft position, L is inductance of the windings (normally negligible), R is resistance of the windings, i is current through the windings, v is the applied voltage, J is the rotor's moment of inertia, b is viscous friction coefficient, τ_D is the output torque, τ_{fric} is friction torque caused by the gearbox. Substituting (3.4) into (3.3) yields:

$$v = \frac{R}{k_t}\left(J\frac{d^2\theta}{dt^2} + b\frac{d\theta}{dt} + \tau_{fric} + \tau_D\right) + k_t\frac{d\theta}{dt} \quad (3.5)$$

Using equation (3.5), the desired control signal v can be calculated from desired actuator torque τ_D, joint velocity $d\theta/dt$, joint acceleration $d^2\theta/dt^2$.

4. Control algorithm

4.1. Feedforward control

If the model of the robot dynamics is exact, and there are no initial state errors, then with the feedforward torque, the robot follows the desired trajectory exactly. The feedforward torque can be calculated as:

$$\tau = M(q)\ddot{q}_D + C(q,\dot{q})\dot{q}_D + G(q) \quad (4.1)$$

where q_D is the vector of desired joint angles. However, because there are always modelling errors and uncertainties, feedforward control should always be used in conjunction with feedback, as discussed next.

4.2. Feedback control

A common feedback controller is linear proportional-integral-derivative control, or PID control. Control input of a PID controller:

$$\tau = K_P(q_D - q) + K_I\int(q_D - q)dt + K_D(\dot{q}_D - \dot{q}) \quad (4.2)$$

where K_P, K_I, K_D are positive gain matrices. Three PID controllers, each for a single joint, are tuned in Simulink and then the simulation result will be used to evaluate the tracking performance of these controllers.

4.3. Feedforward plus feedback control

The passivity-based control input is given by:

$$\tau = M(q)\ddot{q}_D + C(q,\dot{q})\dot{q}_D + G(q) + K_D(\dot{q}_D - \dot{q}) + K_P(q_D - q) \quad (4.3)$$

where K_P, K_D are positive gain matrices. Substituting (4.3) into (2.1) yields:

$$M(q)\ddot{e} + C(q,\dot{q})\dot{e} + K_D\dot{e} + K_P e = 0 \quad (4.4)$$

where $e = q_D - q$ is the tracking error. Define the Lyapunov function as:

$$V = \frac{1}{2}\dot{e}^T M\dot{e} + \frac{1}{2}e^T K_P e \quad (4.5)$$

Its derivative yields:

$$\dot{V} = -\dot{e}^T K_D\dot{e} \quad (4.6)$$

implying that $e \to 0$ as $t \to \infty$. Therefore, the passivity-based controller is stable and will be applied to the manipulator.

4.4. Iterative learning control

ILC algorithms determine the control signal sequence for the subsequent cycle j + 1 [10]:

$$u_{j+1}(k) = u_j(k) + \Delta u_{j+1}(k) \quad (4.7)$$

An additional phase-lead compensation increases the learning bandwidth of the algorithm. The learning control law becomes:

$$u_{j+1}(k) = u_j(k) + \Phi e_j(k+l) \quad (4.8)$$

where $u_{j+1}(k)$ is the control input at time k and iteration $j+1$, Φ is the matrix of learning gains, $e_j(k)$ is the error signal at time k and iteration j, l denotes the phase lead. This control approach needs the adjustment of two parameters: the learning gain Φ and the phase lead l. These adjustments can be achieved experimentally [10].

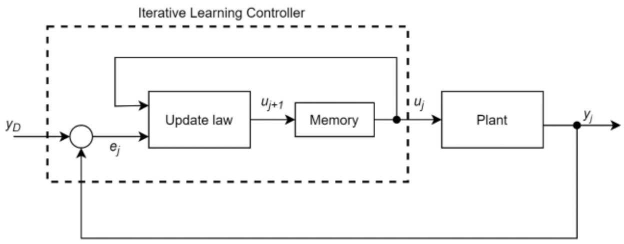

Figure 2 Diagram of ILC

5. Controller simulation

The controller will be simulated while tracking a reference S-Curve trajectory which is smoother than a set point; thus, the robot will operate with reduced vibration and oscillation. The robot dynamics model is imported from Solidworks for simulation. Since the robot is controlled by torque, the three actuated joints will have their z-component torque set as "Provided by Input" and joint angle value set as "Automatically Computed" in the Simulink model.

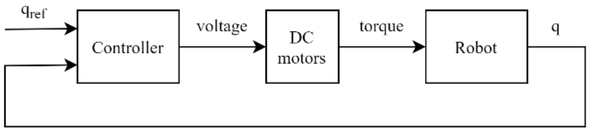

Figure 3 Diagram of the control system

Table 5.1 Parameters of DC motors

Parameter	Motor 1	Motor 2	Motor 3
R (Ω)	2.2472	1.2765	1.8652
J (kg.m²)	0.1645	0.0376	0.0979
b (N.m.s/rad)	0.1883	0.1997	0.3112
k_t *(N.m/A)*	3.3785	3.3510	3.4005
τ_D *(N.m)*	0.3883	0.3962	0.2157

Table 5.2 Parameters of PID controllers

Parameter	Motor 1	Motor 2	Motor 3
K_P	222.7	189.8	210.0
K_I	868.6	741.3	836.4
K_D	2.5	2.2	2.2
t_s *(second)*	0.01	0.01	0.01

The evaluation of the tracking error can be done by mean of RMS error criteria defined for single actuators as:

$$RMSE = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(q_D(k) - q(k))^2} \quad (5.1)$$

and for the complete system as:

$$RMSE = \sum_{i=1}^{3}\left(\sqrt{\frac{1}{N}\sum_{k=1}^{N}(q_{Di}(k) - q_i(k))^2}\right) \quad (5.2)$$

5.1. Feedforward control

From the generated trajectory $(q_D, \dot{q}_D, \ddot{q}_D)$, the required actuator torque for each joint is calculated using Newton-Euler formulation. Using equation (3.5), transform the required torque into required voltage to be applied at each DC motor.

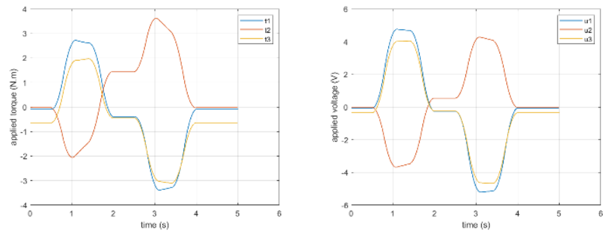

Figure 4 Feedforward torque and voltage

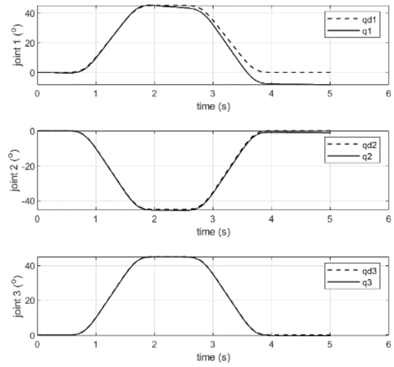

Figure 5 Reference tracking (feedforward control)

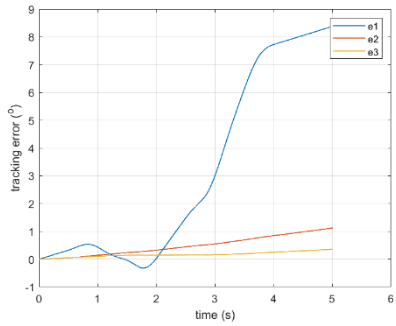

Figure 6 Tracking error (feedforward control)

As can be seen from the figures, the tracking performance of this feedforward controller is not good with large tracking error introduced. Maximum tracking error for joint 1 is almost 9°, while that for joint 2 and 3 are at roughly 1°. The RMS errors are 4.6°, 0.6°, and 0.2°, respectively.

5.2. Feedback control

As can be seen from the figures, the tracking performance of these PID controllers is better compared to that of the feedforward controller. Maximum tracking error for all three joints is at roughly 0.6°. The RMS error is equally 0.3° for all joints.

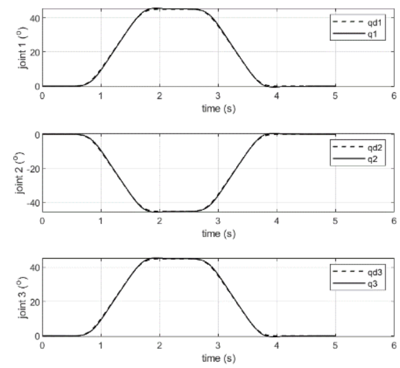

Figure 7 Reference tracking (feedback control)

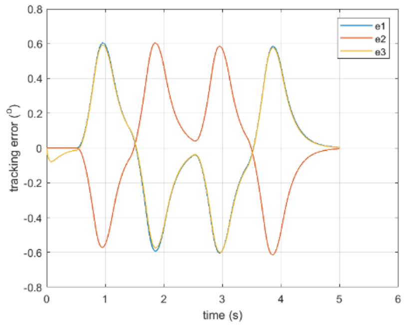

Figure 8 Tracking error (feedback control)

5.3. Feedforward plus feedback control

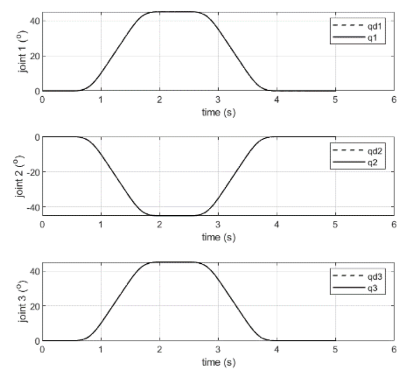

Figure 9 Reference tracking (passivity-based control)

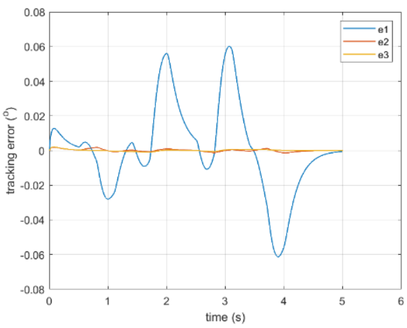

Figure 10 Tracking error (passivity-based control)

As can be seen from the figures, the tracking performance of the passivity-based controller is better compared to that of both the feedforward and feedback control alone. Maximum tracking error for joint 1 is roughly 0.06°, while that for joint 2 and 3 is roughly 0.002°. The RMS errors for three joints are 0.026°, 0.001°, and 0.001°, respectively.

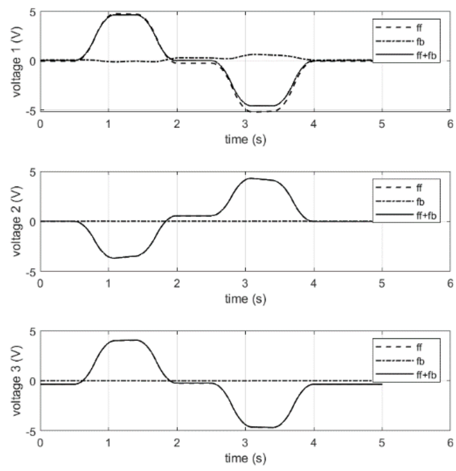

Figure 11 Voltage applied at three motors

Figure 11 shows the control signal supplied to the actuators by the feedforward control alone (ff), feedback control alone (fb), and the total voltage (ff+fb). These plots show typical behavior: the passivity-based controller yields better tracking than either feedforward or feedback alone, with less control effort than feedback alone.

5.4. Iterative learning control

From simulation results, determine the lower and upper bounds of the learning gain $\Phi = [\Phi_1\ \Phi_2\ \Phi_3]^T$. With Φ outside of this range, the control system becomes unstable and tracking error convergence is not guaranteed:

$$0 < \Phi_1, \Phi_2, \Phi_3 \le 7 \qquad (5.3)$$

Similarly, determine the lower and upper bounds of the phase lead index l:

$$0 < l \le 40 \qquad (5.4)$$

Then, for each pair of Φ and l, the control system is simulated for 5 iterations, and the RMS error for the complete system is shown in the figure below.

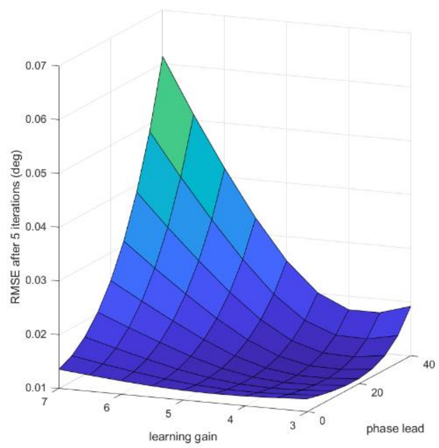

Figure 12 Effect of Φ and l on error convergence

According to simulation result, from an initial RMS error of 0.0274°, the parameters $\Phi = [4.5\ 4.5\ 4.5]^T$ and $l = 10$ yield the best error convergence property with the RMS error after 5 iterations of 0.0105°.

Table 5.3 Parameters for ILC simulation

Parameter	Symbol	Value
Iteration number	n	20
Learning gain	Φ	$[4.5 \quad 4.5 \quad 4.5]^T$
Phase lead index	l	10
Desired RMSE for joint 1	E_{RMS1}	0.0010°
Desired RMSE for joint 2	E_{RMS2}	0.0005°
Desired RMSE for joint 3	E_{RMS3}	0.0005°

The passivity-based controller is then used to simulate the ILC with above parameters. The results are shown in the below figures.

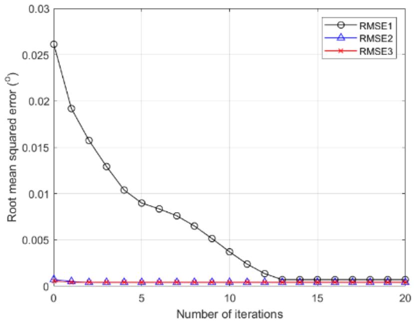

Figure 13 RMS error of three joints after 20 iterations

For joint 1, the RMS error reduced from the initial value of 0.0261° to 0.0007° after 13 iterations. For joint 2, the RMS error reduced from the initial value of 0.0007° to 0.0004° after 2 iterations. For joint 3, the RMS error reduced from the initial value of 0.0005° to 0.0004° after 1 iteration.

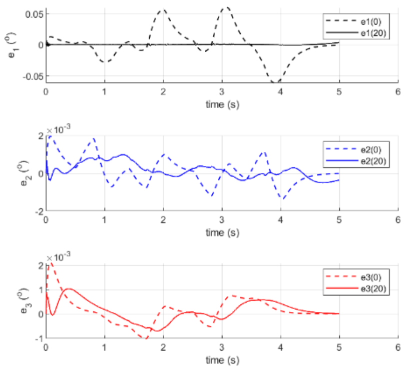

Figure 14 Tracking error (in degree) before and after 20 iterations

6. Conclusion

To increase the accuracy of parallelogram linkage manipulators, Iterative Learning Control (ILC) has been implemented in a control system integrating feedforward and feedback control. The feedforward and feedback control are first designed and tested individually, and later integrated in the same system. The results are very promising since all algorithms yield an important improvement of accuracy and a bettering of tracking performance. The use of additional learning techniques has allowed the application of simple and efficient linear learning methods.

Acknowledgement: This research is funded by Office for International Study Programs (OISP), Ho Chi Minh City University of Technology (HCMUT), VNU-HCM under grant number **(SVOISP-2021-CK-45).** We acknowledge the support of time and facilities from HCMUT, VNU-HCM for this study.

7. References

[1] Mark W. Spong, Seth Hutchinson, and M. Vidyasagar (2020), *Robot Modeling and Control - 2nd Edition*, John Wiley & Sons, Ltd.

[2] K. M. Lynch and F. C. Park (2017), *Modern Robotics*, Cambridge University Press.

[3] Luc Jaulin (2015), *Automation for Robotics*, Wiley.

[4] Haugen, F. (2009). *Basic Dynamics and Control*.

[5] Nan Luan Haiqing Zhang, and Shanggao Tong, (2012), "*Optimum motion control of palletizing robots based on iterative learning*", Industrial Robot: An International Journal, Vol. 39 Iss 2 pp. 162 - 168

[6] Svante Gunnarsson, Mikael Norrlof, (1997), *A Short Introduction to Iterative Learning Control.*

[7] Paul, Richard (1981). *Robot manipulators: mathematics, programming, and control: the computer control of robot manipulators.* MIT Press, Cambridge, MA. ISBN 978-0-262-16082-7.

[8] Craig, 1955 (1986), *Introduction to Robotics: Mechanics & Control*, Addison-Wesley Pub. Co., ISBN: 0201103265.

[9] B. Siciliano and O. Khatib (2007), *Handbook of Robotics*, Springer-Verlag, Berlin, Heidelberg.

[10] H. Abdellatif, M. Feldt and B. Heimann, *"Application study on Iterative Learning Control of high speed motions for parallel robotic manipulator,"* 2006 IEEE Conference on Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control, 2006, pp. 2528-2533, doi: 10.1109/CACSD-CCA-ISIC.2006.4777036.