Multi-robot Cooperative System Modeling and Control Software Development

Tianhong Cheng, Zhiwei Wu, and Wenfu Xu(✉)

School of Mechanical Engineering and Automation, Harbin Institute of Technology Shenzhen,
Shenzhen 518055, China
chengtianhong@stu.hit.edu.cn, wfxu@hit.edu.cn

Abstract. In order to solve the problem that it is difficult to achieve better multi-robot coordination operation using the original control system in multi-robot cooperative systems, this paper conducts system modeling, control framework design and software development and verification experiments for multi-robot cooperative systems. We firstly model the system composed of multiple robots and operated objects. The coordination between any two robots can be divided into two categories: loose coordination and tight coordination according to the constraint relationship between the robots and the objects, based on which the kinematics and dynamics analysis of the multi-robot system is carried out. A multi-robot cooperative control framework is designed, which contains modules for task planning, force and position assignment, sensor information acquisition, position/force control and synchronization clock. On the basis of the multi-robot cooperative control system, the robot cooperative control software is designed and developed, and the software system contains, with excellent characteristics. Finally, an experimental environment containing three large-load EFORT industrial robots was established and experiments were conducted using the developed multi-robot cooperative control software to verify the usability and reliability of the software system.

Keywords: Multi-robots cooperative system · Control software development · Robot control

1 Introduction

Collaborative multi-robot operation is the focus of current robotics applications, especially in industrial robots, where multi-robot collaboration has the advantages of large loads, high efficiency, and higher flexibility and organization to better meet the demands of increasingly complex applications [1, 2].

However, in current industrial robot applications, control is usually performed through the control system accompanying the robot, where each robot corresponds to a dedicated set of control devices. And the independent motion of each robot is achieved by operating and programming each controller separately. For systems with multiple robots, which often do not truly achieve the characteristics of synergy in time and space [3]. This control method neither ensures the synchronization of robot movements nor

X.-J. Liu et al. (Eds.): ICIRA 2021, LNAI 13014, pp. 14–24, 2021.
https://doi.org/10.1007/978-3-030-89098-8_2

enables fast information exchange between robots, which makes it impossible for multiple robots to perform the same task simultaneously in an overlapping operation space, and at the same time. Each robot has to be operated or programmed separately for each robot, which greatly increases the difficulty and complexity of implementing a multi-robot cooperative operating system.

The common controller architectures are "PC + motion control card" [4] and "PC + real-time operating system + industrial bus" [5, 6]. The EtherCAT bus is becoming a hot spot for controller research [7, 8].

Traditional controllers only support position control, which has large limitations in compliance [9]. Commonly used compliance control methods include impedance control, force/position hybrid control, etc. [10]. And coordinated compliance control is a concern in multi-robot coordinated motion [11–13].

The main objective of this paper is to propose a general multi-robot cooperative control framework and design a scalable, easy-to-use, and high-performing multi-robot cooperative control software based on this.

2 Multi-robot Collaborative System Modeling

2.1 Coordinated Operation Model

Modeling and kinematic and dynamical analysis of multi-arm robots are the basis for studying multi-robot cooperative control systems. Generally, multi-robot systems consist of a base, robots and operated objects, and for the basic two-arm cooperative system, there are two cases of tight coordination and loose coordination between two robots according to the kinematic and dynamical constraint relationships between robots.

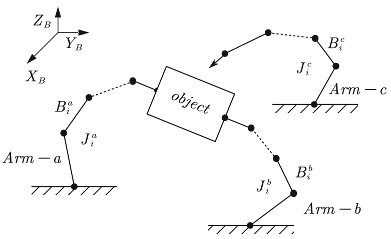


Fig. 1. Multi-arm collaborative system configuration

And for the case of more robots with the same operated operation, a tight coordination relationship is presented between robots with a fixed constraint with the same operated object, and a loose coordination relationship is presented between robots without a constraint relationship. Consider a multi-robot cooperative system consisting of three robots and an operated object, three robots with the same object (see Fig. 1), assuming that robot A, robot B have constraint relationship with the object and robot C has no constraint relationship with the object, then robot A has a tight coordination relationship with robot B and has a loose coordination relationship with robot C.

2.2 Kinematic Modeling

For a single robot, the positional relationship between its position and the joint vector can be expressed by Eq. 1.

$$ {}^{0i}\boldsymbol{T}_{ei} = {}^{0}\boldsymbol{T}_{e}^{i} = \mathrm{fkine}\left(\boldsymbol{q}_i\right) \tag{1} $$

The relationship between end velocity and joint angular velocity can be expressed by the Eq. 2.

$$ \begin{bmatrix} \boldsymbol{v}_e^i \\ \boldsymbol{\omega}_e^i \end{bmatrix} = \boldsymbol{J}_i\left(\boldsymbol{q}_i\right)\dot{\boldsymbol{q}}_i \tag{2} $$

For the case where there are constraints between the robot and the object can be expressed by the Eq. 3.

$$ {}^{0i}\boldsymbol{T}_{ei} = \left({}^{B}\boldsymbol{T}_{0i}\right)^{-1} \cdot {}^{B}\boldsymbol{T}_{L} \cdot \left({}^{ei}\boldsymbol{T}_{L}\right)^{-1} \tag{3} $$

The velocity of the object is analyzed as shown in the Fig. 2 and satisfies the Eq. 4.

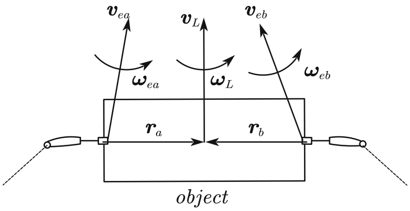


Fig. 2. Object velocity analysis

$$ \begin{bmatrix} \boldsymbol{v}_L \\ \boldsymbol{\omega}_L \end{bmatrix} = \begin{bmatrix} \boldsymbol{E} & -\left(\boldsymbol{r}_i\right)^{\times} \\ \boldsymbol{0} & \boldsymbol{E} \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_{ei} \\ \boldsymbol{\omega}_{ei} \end{bmatrix} \tag{4} $$

For robots with constraints between them and the object, the end velocity is the same as the object.

$$ \dot{\boldsymbol{q}}_i = \boldsymbol{J}_i^{\mathrm{T}}\left(\boldsymbol{q}_i\right) \begin{bmatrix} \boldsymbol{E} & -\left(\boldsymbol{r}_i\right)^{\times} \\ \boldsymbol{0} & \boldsymbol{E} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{v}_L \\ \boldsymbol{\omega}_L \end{bmatrix} \tag{5} $$

2.3 Dynamics Modeling

For each robots.

$$ \boldsymbol{M}_i\ddot{\boldsymbol{q}}_i + \boldsymbol{C}_i\left(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i\right)\dot{\boldsymbol{q}}_i + \boldsymbol{G}_i\left(\boldsymbol{q}\right) = \boldsymbol{\tau}_{li} + \boldsymbol{\tau}_{exti} \tag{6} $$

The joint torque and the robot end-action force are expressed by Eq. 7.

$$\boldsymbol{\tau}_{exti} = \boldsymbol{J}_i^T \boldsymbol{F}_{\mathrm{e}i} = \boldsymbol{J}_i^T \begin{bmatrix} \boldsymbol{f}_{\mathrm{e}i} \\ \boldsymbol{\tau}_{\mathrm{e}i} \end{bmatrix} \tag{7}$$

The object is subjected to the robot end-action and environmental forces that are bound to its existence, which can be expressed as Eq. 8.

$$\begin{aligned} &\begin{bmatrix} -\boldsymbol{G}_L \\ \boldsymbol{0} \end{bmatrix} + \begin{bmatrix} \mathrm{m}_L \dot{\boldsymbol{v}}_L \\ \boldsymbol{I}_L \dot{\boldsymbol{\omega}}_L + \boldsymbol{\omega}_L \times (\boldsymbol{I}_L \boldsymbol{\omega}_L) \end{bmatrix} \\ &= \sum \begin{bmatrix} \boldsymbol{E} & \boldsymbol{0} \\ \boldsymbol{r}_i^{\times} & \boldsymbol{E} \end{bmatrix} \begin{bmatrix} -\boldsymbol{f}_{ei} \\ -\boldsymbol{\tau}_{ei} \end{bmatrix} - \begin{bmatrix} \boldsymbol{E} & \boldsymbol{0} \\ \boldsymbol{r}_L^{\times} & \boldsymbol{E} \end{bmatrix} \begin{bmatrix} -\boldsymbol{f}_L \\ -\boldsymbol{\tau}_L \end{bmatrix} \end{aligned} \tag{8}$$

3 Collaborative Control System Frame and Methods

3.1 Control Frame Design

The controller system mainly consists of task planning, machine force/position controller, and sensor information acquisition.

Task planning acquires the position and force planning of the robot, performs position and force assignment. And sends it to the robot's force controller or position controller, which outputs the desired joint trajectory, sends it to the robot and executes it (Fig. 3).

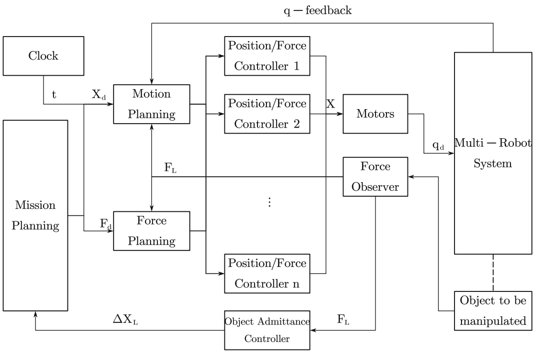


Fig. 3. Overall diagram of the coordinated control algorithms.

3.2 Position and Force Distribution

The case where there are two robots exerting forces on the object can be expressed by the Eq. 9.

$$\boldsymbol{F}_{bed} = \boldsymbol{\Gamma}_{eb}^{-1} \left(-\boldsymbol{\Gamma}_{ea} \boldsymbol{F}_{ea} + \boldsymbol{\Gamma}_L \boldsymbol{F}_L + \begin{bmatrix} \boldsymbol{G}_L \\ \boldsymbol{0} \end{bmatrix} - \begin{bmatrix} \dot{\boldsymbol{P}}_L \\ \dot{\boldsymbol{L}}_L \end{bmatrix} \right) \tag{9}$$

With the task trajectory, the robot planning trajectory with constraint relationship with the object can be found.

$$ {}^{0i}\boldsymbol{T}_{ei} = {}^{0i}\boldsymbol{T}_{B} \cdot {}^{B}\boldsymbol{T}_{L} \cdot {}^{L}\boldsymbol{T}_{ei} \tag{10} $$

3.3 Motor Control and Motion Planning

The multi-robot cooperative control system designed in this paper is mainly for industrial robots with large mass and inertia, and a primary issue is to ensure smooth and continuous joint motion in position, velocity, and acceleration. The servo motors used in general robots can be controlled relatively easily in terms of position or velocity. In order to reduce the vibration caused by velocity changes, the control of a single motor axis is achieved by applying PD control on the motor velocity loop.

In order to ensure the synchronization of multi-axis and multi-robot position motion, taking into account the smoothness of motion and efficiency requirements, a five times spline method is used for planning each motor axis.

$$ \begin{cases} \theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \\ \dot{\theta}(t) = a_1 + 2a_2 t + 3a_3 t^2 + 4a_4 t^3 + 5a_5 t^4 \\ \ddot{\theta}(t) = a_2 + 6a_3 t + 12a_4 t^2 + 20a_5 t^3 \end{cases} \tag{11} $$

The vector of coefficients can be obtained by the method of coefficients to be determined.

$$ [a_i] = f\left(\theta_0, \theta_f, \dot{\theta}_0, \dot{\theta}_f, \ddot{\theta}_0, \ddot{\theta}_f, t_f\right) \tag{12} $$

The initial value is the current value, the end position is given, the end velocity and acceleration are set to 0. In order to achieve multi-axis synchronization, a maximum average velocity of v is set, and the estimated time for each axis to move to the current command is given by Eq. 13.

$$ [t_i] = ([\theta_{if}] - [\theta_{i0}])/v \tag{13} $$

Then the planning time is given by Eq. 14.

$$ t_f = \max(t_i) \tag{14} $$

3.4 Multi-arm Compliance Control Method

In the actual application of robot, the pure position control cannot meet the requirement of flexibility, and it is easy to cause damage to robot or object. For the robot without constraint, the traditional impedance control is used, and for the robot with object with constraint, the main concern is the compliance control between object and environment. A dual closed-loop impedance controller is used to perform force control between robot/object and object/environment at the same time (Fig. 4).

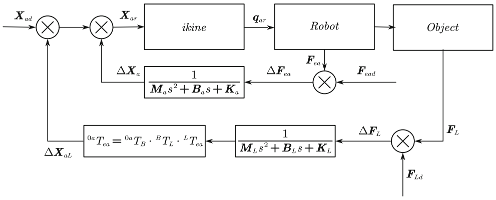

X_{ad}
X_{ar}
ikine
q_{ar}
Robot
Object
F_{ea}
ΔX_a
$\frac{1}{M_a s^2 + B_a s + K_a}$
ΔF_{ea}
F_{ead}
F_L
ΔF_L
${}^{0a}T_{ea} = {}^{0a}T_B \cdot {}^{B}T_L \cdot {}^{L}T_{ea}$
$\frac{1}{M_L s^2 + B_L s + K_L}$
ΔX_{aL}
F_{Ld}

Fig. 4. Object-based impedance controller

4 Control System Software

4.1 Demand Analysis

The control system software has three main requirements.

Scalability. Heterogeneous collaboration of multiple robots is possible, compatible with existing industrial solutions, and allows for rapid expansion of multi-robot systems.

Ease of use. Provides good interfaces at different levels to facilitate the expansion of operating environments, algorithm development and verification.

High performance. Meet real-time requirements, provide multiple computing unit expansion interfaces, and fully mobilize the resources of self-developed hardware.

4.2 Overall Design

The entire software system is divided into five main layers, namely user interface layer, task planning layer, controller layer, device management layer, and hardware interface layer (Fig. 5).

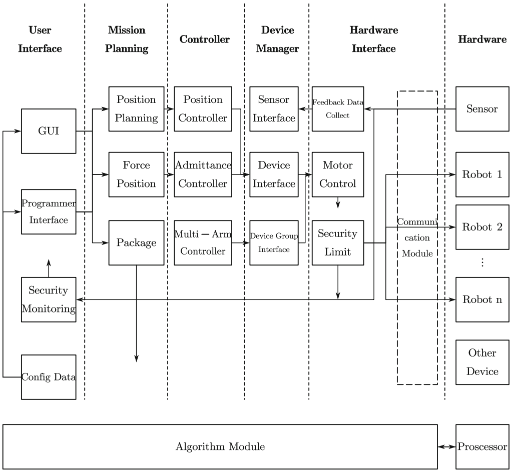


Fig. 5. Overall software frame design

4.3 Design of Each Layer

The user interface layer contains modules such as graphical user interface, programming interface, and safety monitoring for direct interaction with the user.

Task planning layer. The task planning layer integrates the lower level position/force controllers, process equipment packages, and unified scheduling through Python scripts, and can integrate ROS and Python rich resources.

The controller layer implements advanced control algorithms, including impedance control, Cartesian space trajectory control, and multi-robot cooperative control algorithms, accepts feedback data from the lower layer and sends commands to the device or device group, facilitating the development and application of advanced algorithms.

Device management layer. The device interface layer integrates the basic control units of the lower layer into independent devices or groups of devices from multiple devices for dynamic management of execution and sensing devices, facilitating the application of complex coordination algorithms for multiple robots to a specific set of devices for easy development and use.

The hardware interface layer communicates directly with the underlying hardware, performs motor axis motion planning through multi-axis synchronization algorithms, and accepts feedback from encoders and other sensors, and is compatible with a variety of communication modules, enabling the connection of different kinds of devices in industrial solutions.

The hardware environment can include robots, sensors, process equipment, etc.

4.4 Main Modules

Some important modules of the software system are described.

Communication Module. In order to synchronize the controller clock with the EtherCAT reference clock, the reference clock is used as input and the PD control is used to correct the controller clock.

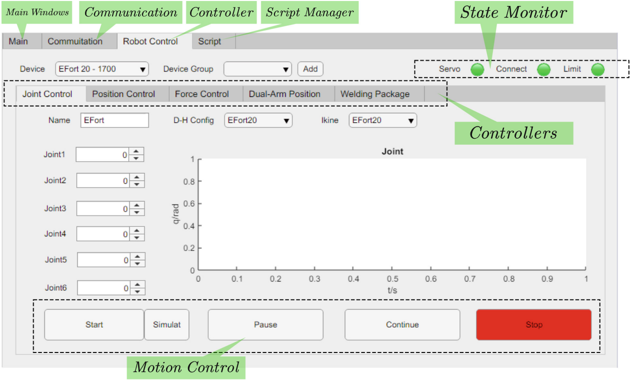
Calculation Module. In order to ensure the performance and stability of the controller, a coprocessor is used for the design of the operation library. Since often the same algorithm is operated for multiple axes or devices, parallel methods can be utilized and dynamic load optimization and redundant operations are implemented.

Safety Monitoring. The safety mechanism consists mainly of the underlying position, speed and force limits and state machine switching. With EtherCAT the current position, speed and current can be quickly obtained, the limits are exceeded as Eq. 15 and a state machine switch is triggered to ensure safety.

$$\begin{cases} \min(q_i) < q < \max(q_i) \\ |v_i| < v_{i\max} \\ T|i_i| < \tau_{i\max} \end{cases} \tag{15}$$

4.5 Software Interface

The graphical user interface contains the main functions of the force control system. It includes communication module, robot control, package usage, script management and running (Fig. 6).



Device EFort 20 - 1700

Name EFort

D-H Config EFort20

Ikine EFort20

Fig. 6. Graphical user interface

5 Experiment Verification

5.1 Experimental System

The experimental environment was set up with three Efort ER20-1700 industrial robots, all connected to the control system software using EtherCAT.

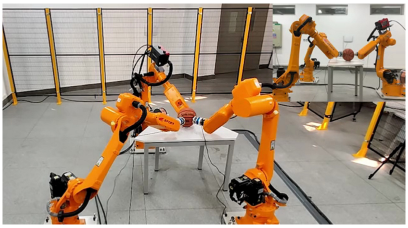


Fig. 7. Experimental System

5.2 Motion Planning Experiment

The three robots are kept in a constant relative position to each other and move linearly along the x-axis in space to obtain the actual robot end x-axis direction position and compare it with the planned position (Figs. 8 and 9).

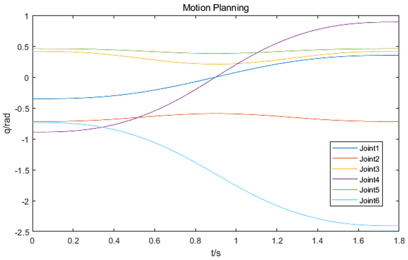


Fig. 8. Planning trajectories of joints

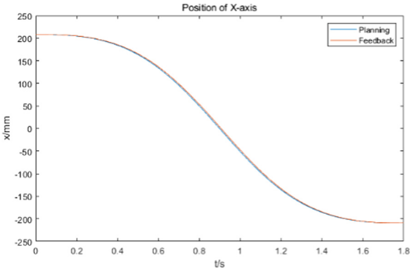


Fig. 9. Actual end x-axis position of robot A

The figure represents the joint trajectory planning of robot A and the actual end x-axis trajectory, and it can be seen that the minimum trajectory error is 0.31 mm, which has sufficient accuracy.

5.3 Compliance Control Experiment

Make three robots hold a basketball at an angle of 120° as Fig. 7, one robot is position controlled, and the other two robots are impedance controlled in the direction of the end z-axis. Make the basketball gradually rises, and obtain the feedback value of the robot end sensor to prove that this multi-robot motion control system is capable of complex force control algorithms (Fig. 10).

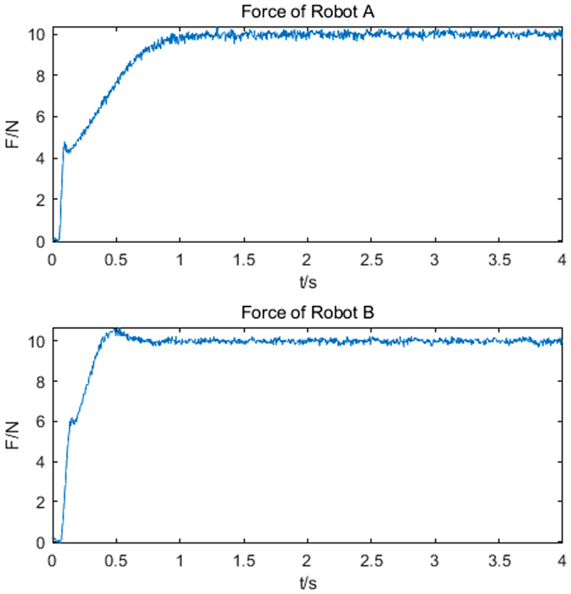


Fig. 10. Robot A and Robot B force sensor feedback

The figure shows the end force sensor feedback values for Robot A and Robot B using force control with a desired force of 10 N. It can be seen that the force control accuracy of the impedance controller is less than 1%.

6 Conclusion

In this paper, we concern the problem of multi-robot cooperative control. Firstly, a basic multi-robot operating system model is established, and on the basis of the model, kinematics and dynamics are analyzed for the two cases of constrained and unconstrained between robots and objects, respectively. Then the multi-robot cooperative motion control system framework is designed, the force and position distribution methods are given, the single-axis motor control and multi-axis synchronous motion planning methods are designed. And the multi-robot cooperative soft-smooth control algorithm based on object impedance is given. Based on the control framework, the multi-robot cooperative control system software is designed and developed. The software system has hardware interface layer, device management layer, control layer, task layer and user layer, and contains communication module, scalable computing module, safety monitoring and fault handling module, with strong scalability, ease of use and usability. Finally, an experimental environment containing three industrial robots was built, and experiments on multi-robot trajectory motion and multi-robot coordinated impedance control were conducted with good results to verify the usability of the control system software.

Acknowledgement. This work was supported by the Key-area Research and Development Program of Guangdong Province [grant numbers 2019B090915001].

In addition, the authors would like to acknowledge the following individuals for their contributions by providing technical insight and guidance to the authors.

Bowen Wang, Master. Harbin Institute of Technology Shenzhen.

Tao Chen, Master. Harbin Institute of Technology Shenzhen.

References

1. Verma, J.K., Ranga, V.: Multi-robot coordination analysis, taxonomy, challenges and future scope. J. Intell. Rob. Syst. **102**(1), 1–36 (2021). https://doi.org/10.1007/s10846-021-01378-2
2. Saïdi, F., Pradel, G.: Contribution to human multi-robot system interaction application to a multi-robot mission editor. J. Intell. Rob. Syst. **45**, 343–368 (2006)
3. Zhuang, Y., et al.: Multi-robot cooperative localization based on autonomous motion state estimation and laser data interaction. Sci. China Inf. Sci. **53**, 2240–2250 (2010)
4. Farooq, M., Wang, D.: Implementation of a new PC based controller for a PUMA robot. J. Zhejiang Univ. Sci. A **8**(12), 1962–1970 (2007). https://doi.org/10.1631/jzus.2007.A1962
5. de Oliveira, R.W.S.M., et al.: A robot architecture for outdoor competitions. J. Intell. Rob. Syst. **99**(3–4), 629–646 (2020). https://doi.org/10.1007/s10846-019-01140-9
6. Han, S.J., Oh, S.Y.: An optimized modular neural network controller based on environment classification and selective sensor usage for mobile robot reactive navigation. Neural Comput. Appl. **17**, 161–173 (2008). https://doi.org/10.1007/s00521-006-0079-1
7. Aristova, N.I.: Ethernet in industrial automation: Overcoming obstacles. Autom. Remote. Control. **77**(5), 881–894 (2016). https://doi.org/10.1134/S0005117916050118
8. Huang, R.-Y., Chen, Y.-J., Chen, Y.-X., Cheng, C.-W., Tsai, M.-C., Lee, A.-C.: Advanced application of centralized control for a scanning mirror system based on etherCAT fieldbus. Int. J. Control Autom. Syst. **19**(3), 1205–1214 (2021). https://doi.org/10.1007/s12555-019-0754-5
9. Xu, G., Song, A., Li, H.: Adaptive impedance control for upper-limb rehabilitation robot using evolutionary dynamic recurrent fuzzy neural network. J. Intell. Rob. Syst. **62**, 501–525 (2011)
10. Furtado, G.P., Americano, P.P., Forner-Cordero, A.: Impedance control as an optimal control problem: a novel formulation of impedance controllers as a subcase of optimal control. J. Braz. Soc. Mech. Sci. Eng. **42**(10), 1–20 (2020). https://doi.org/10.1007/s40430-020-02586-x
11. Vukobratović, M.K., Rodić, A.G., Ekalo, Y.: Impedance control as a particular case of the unified approach to the control of robots interacting with a dynamic known environment. J. Intell. Rob. Syst. **18**, 191–204 (1997). https://doi.org/10.1023/A:1007915307723
12. Dubowsky, S., Sunada, C., Mavroidis, C.: Coordinated motion and force control of multi-limbed robotic systems. Auton. Rob. **6**, 7–20 (1999). https://doi.org/10.1023/A:1008816424504
13. Yan, L., Xu, W., Hu, Z., Liang, B.: Virtual-base modeling and coordinated control of a dual-arm space robot for target capturing and manipulation. Multibody Sys. Dyn. **45**(4), 431–455 (2018). https://doi.org/10.1007/s11044-018-09647-z