ATLANTIS PRESS
Advances in Intelligent Systems Research (AISR), volume 151
2018 International Conference on Computer Modeling, Simulation and Algorithm (CMSA 2018)

Multi-robot Synchronous Control Based on Multi-thread

Peng Xiong, Bitao Long, Zhengfu Lu, Xuwu Liu and Yulian Jiang*
College of Electrical and Information Engineering, Southwest Minzu University, Chengdu, Sichuan, China
*Corresponding author

***Abstract*—With the development of robot technology, many researchers have changed the investigating direction from single-robot control to multi-robot control. In terms of the group control of humanoid robot, this paper addressed a socket communication and multi-thread technology based on Python such that the actions of multiple robots could be synchronously controll. It could be proved by experiments that the delay of starting dancing program of robots could be reduced to milliseconds by applying the method and the dancing actions of multiple robots were consistent from eyes observation, which leaded to the actions synchronization of multiple robots. Furthermore, the successful rate of many real machine experiments reached 94.5%, which verified the effectiveness of the proposed method.**

Keywords—robot; socket communication; multi-thread; synchronous control; choregraphe software

I. INTRODUCTION

With the rapid development of robot technology, the French company, Aldebaran Robotics, has successful developed an intelligent humanoid robot called NAO. NAO has been purchased by many colleges and it was used for the study of multiagent, automatic control, signal processing and route plan and has now been widely used humanoid robot in the academic field. From the fully autonomous collective dance of 20 NAO robots at the World Exposition in 2012 to the wonderful collective performance of 540 Alpha 1S robots at the Spring Festival Gala Evening in 2016, all performance involved the issue of multi-robot synchronous control. Speech recognition control and Choregraphe software control are the commonly synchronous control method, in which multiple NAO robots start the dancing program by recognizing the sound signal for the former, but this method does not guarantee that all robots can always be synchronized and the main reason is that it is hard to keep the moment of recognizing sound consistent for each robot; for the later, based on Choregraphe, a one-on-one control can be executed for the NAO robots, that is, each NAO robot is connected with a PC by Choregraphe, each PC is operated by a corresponding operator and all operators simultaneously click the starting button of acting program[1][2]. However, for different PC systems, the time required for the robot to complete the same program can not be exactly the same. So it is quite difficult to keep the consistent action at the same moment.

In order to overcome above mentioned disadvantages, this paper addresses a method of multi-thread-based multi-robot synchronous control[3]. Its core lies in the following aspects: One of the robots is set as PC server, other robots are set as PC client and each client is created a sub-thread object; when the server recognize the speech order sent by operators, the program of sending message is executed and the message that can be set by ourselves is the content of the socket; then the rest robots of client will receive the order message and act simultaneously. It can be proved by experiments that multiple robots can be effectively controlled to perform the dance action synchronously and the error can be reduced to milliseconds, that is, we do not distinguish the differences by eyes.

II. THE HARDWARE AND CONFIGURATION OF NAO ROBOT

NAO is a smart biped robot with 25 degrees of freedom (*DOF*), 2 cameras at the head, 4 microphones, 1 ultrasonic distance sensor, 2 infrared transmitters and receivers. Besides, it is also equipped with a series of touch sensors, Hall sensors, pressure sensors, impactors, sonar and inertial sensing unit throughout the whole body.

The NAO robot has two CPUs. One is on the head and the other is on the trunk, where the internal 1.6GHz CPU handles the entire robot's activity information. Moreover, NAO has the ability of randomly accessing memory of 1GB and external storage of 2GB and all software is controlled by a dedicated operating system NAOqi. Choregraphe software is a programming software developed by Aldebaran Corporation and it can be used for writing program, downloading programs, simulation, detecting the real-time data of robot. This paper uses Python programming language. Based on the Python development environment (*PyCharm*), the right debug program can be imported into Choregraphe's box and we can run the program of box to control the movement of the NAO robot.

III. MULTI-ROBOT SYNCHRONIZATION CONTROL

In this paper, based on the synchronization control method of socket multi-threading technology, it can be completed to control the synchronized collective dance of three NAO robots.

A. Multi-robot Lan Formation

NAO robots have two kinds of network connection, wired and wireless connection. This paper uses the wireless connection under the same local area network such that operators can control the NAO robot to execute the program under the same local area network and multiple robots can communicate between networks. A router is set as a node in

this paper and each robot is considered as a stand-alone PC. It can be connected with the router by WIFI and then multiple robots are automatically assigned IP addresses with the same network number (*under the same LAN*), which provides the basic communication for us to use the platform of choregraphe software to control synchronous movement of multiple robots[4].

B. Speech Recognition

The ALSpeechRecognition Module of NAO robots is used to identify a word or phrase of predefined language[5][6]. We set the predefined language to simplified Chinese and the phrase list for voice matching. After the program used for recognition is started, the ALSpeechRecognition is placed in the key value SpeechDetected. It is a Boolean value that is used to detect if someone sounds. When the sound is detected, the speech recognition module of the NAO robot converts the speech signal into a series of strings and compares the string with the elements in the list of phrases set previously and then places the best matching element in the keyword WordRecognized. Inside of the speech recognition system of the NAO robot has a confidence threshold. If the recognition result is lower than the threshold set by us, the speech recognition fails. Therefore, the successful rate of the speech recognition is affected by ambient noise.

C. Synchronous Control

In this paper, TCP-based multi-threaded concurrent communication of Socket is used to make the actions of robots synchronous. Communication between different computers achieves through the network socket (*Socket*), in which socket exists between the application layer and the transport layer of the TCP / IP protocol[7][8]. Generally, the server is in an endless loop waiting for the client to connect. Socket is the client and server-side communication and the socket object are established in the client and server. The data transmission is completed by the socket object, which ensures the stable network communication between computers[9][10]. The running principle is shown in Figure I.

Multi-threading technology is a technology that implements concurrent execution of multiple threads from software or hardware, which leads to the running of multiple tasks and satisfies exactly the control requirement of synchronous action of three robots in this paper[11][12]. The program of dance action executed concurrently for a number of NAO robots is considered as a number of tasks. Combining with socket communication and multi-threaded method can be used to start above tasks simultaneously.

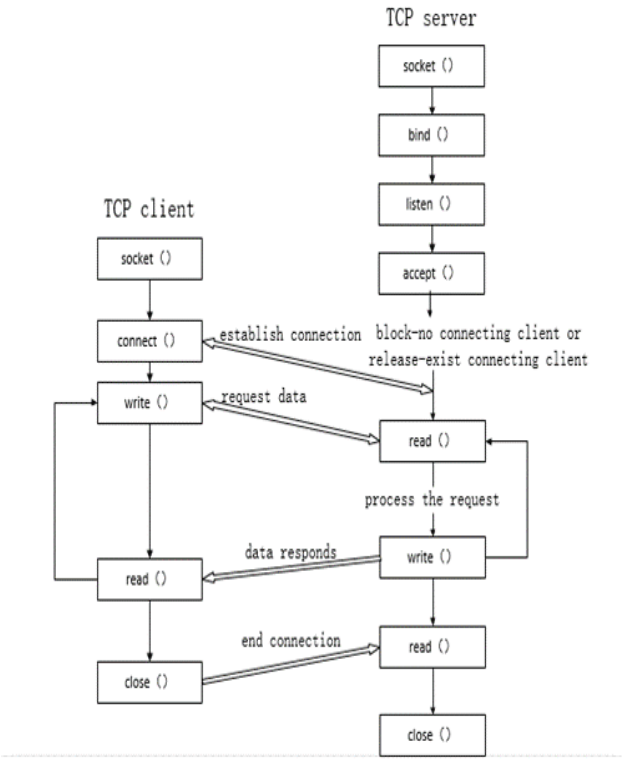


FIGURE I. THE DIAGRAM OF SOCKET RUNNING PRINCIPLE

The flow chart of this method is shown in Figure II. The sounds signal set by operator can be set as a trigger for controlling the synchronization of actions[13]. NAO1, NAO2 and NAO3 represent the number of three robots, respectively, in which NAO1 is set as the server side, NAO2 and NAO3 are set as the client side. NAO1 receives the operator's voice command. If the voice command can be recognized, NAO1 will trigger the output point of self.onStopped( ) method of Speech Reco voice recognition command box, execute the self-prepared Server_Send program of command box server. After that, the program enters the blocking and waits for the client connection and for each connection, the client creates a thread. Then the server side sends information to the client in parallel. After multi-robot client receives the information, the output point of self.onStopped( ) method is triggered and in order to achieve synchronization of the dance movement at the same time, the command box program of actions is executed[14].

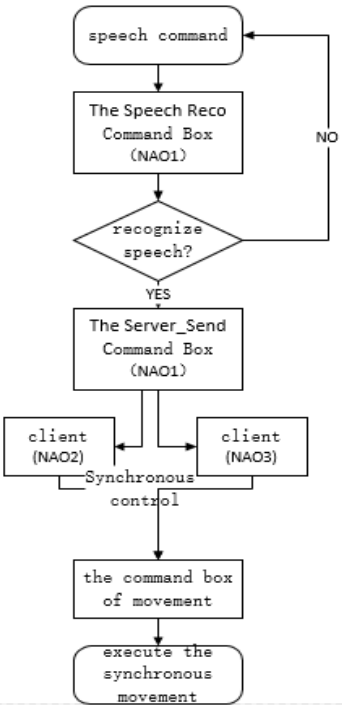


FIGURE II. THE CONTROLLING FLOW CHART OF MULTI-ROBOT SYNCHRONIZATION

Where the process of receiving information simultaneously for each NAO client is shown in Figure III.

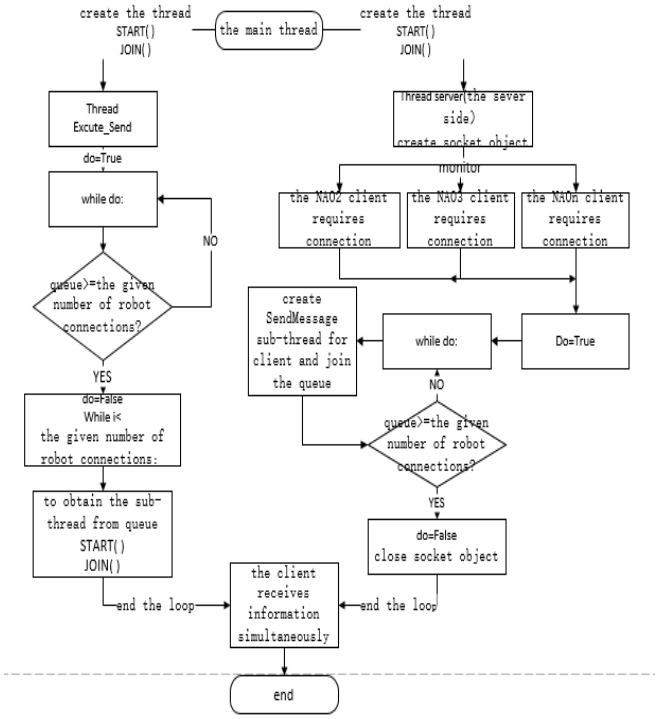


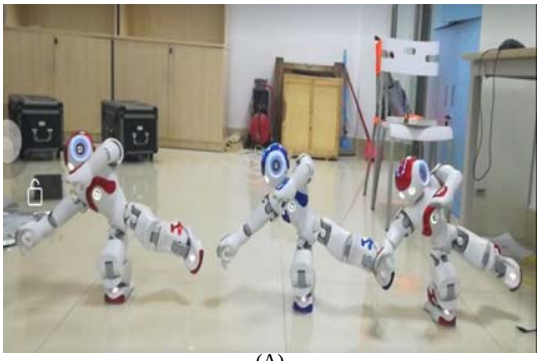
FIGURE III. THE CONTROLLING FLOW CHART OF MULTI-ROBOT SYNCHRONIZATION

Because the two robot clients NAO2, NAO3 need to connect with NAO1 server, variable value of robot_client_num is set as 2 and considered as the symbol of number of clients, which can be used as a loop condition to be determined. The server creates three functions, Server, Excute_Send and SendMessage. The SendMessage function is used for realizing the socket.send( ) sending message to the client. Server function creates a socket object, sets the port reusable, binds the address port, enters the connection status of the monitor and waiting, and sets the while loop for creating an executive function thread SendMessage for each connected client socket. Then the thread is placed in the executive queue. When the queue length equals robot_client_num, the loop end and the server socket object is closesd, that is, the thread has been terminated. Excute_Send function can determine whether the length of the queue is equal to the value of the robot_client_num variable. If the two values are equal, we use the loop while to get the socket thread saved in the queue and execute the start( ) and join( ) functions. The purpose of adding the join( ) function lies in the following point: When the sub-thread ends, the parent thread's program can be ran; if not, the program then enters the blocking state of loop waiting. The main thread creates two function execution threads as server = threading.Thread (*target = Server, args = (robot_client_num*)) and excute_send = threading.Thread *(target = Excute_Send, args = (obot_client_num))* and the two functions *(start( ) and join( ))* are executed. The join ( ) function here is used to wait for the end of thread server and excute_send. Then the self.onStopped( ) method of the command box is executed to control the server-side robot NAO1 and the client robots NAO2 and NAO3 that have received the information to start the action program synchronously.

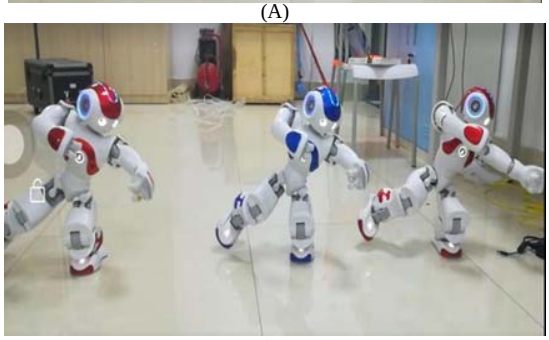
Because thread server and excute_send shares queue resources and there exist the mutex and condition scalar in the queue, the use of python queue can ensure the safety of thread. When the operator issues a voice command to NAO1, NAO1 recognizes the signal and processes it quickly, that is, the server-side server_send command box program is started and then runs first the command box program of client NAO2, which makes the program enter the blocked state. Then the program of client NAO3 is ran and NAO2 and NAO3 will receive the message at the same time and call the output point of self.onStopped( ) method in the command box such that NAO1, NAO2 and NAO3 simultaneously activate the action command box, which realizes the synchronized dance of multiple NAO robots. In this paper, the millisecond datatime module is used and it has been detected that the time of receiving information from the server to client is identical[15][16].

D. Experiments

In this paper, based on the synchronization control method of socket multi-threaded technology, it can be completed to make the dance actions of three NAO robots synchronized. It is shown in Figure IV.



(A)



(B)

FIGURE IV. THE EXPERIMENT OF THREE ROBOTS

IV. CONCLUSION

The problem of synchronous control is one of the researching hotspots of multi-robot control. Based on the control platform of choregraphe software, this paper addresses socket communication and multi-threading technology taking the NAO robot as the research object. The control thread of multiple NAO robots can be started by using voice recognition signal to a NAO robot and then the dance program is started simultaneously, which realizes the synchronous control of multi-robot motion. The experiments show that the method of using socket communication and multi-threading technology to set the client and server greatly reduces the multi-robot control delay and effectively realizes the synchronization action of multiple NAO robots.

ACKNOWLEDGMENT

This work was supported by the Fundamental Research Funds for the Central Universities, Southwest Minzu University (*NO.2016NZYQN07*).

REFERENCES

[1] R A Brooks, “A robust layered control system for a mobile robot,” International Journal of Robotics and Automation, 2(1), pp. 14-23, 1986.

[2] X Wan，X Tian and W Wang, “Application of Speech Recognition Technology in Robot Control, ” Journal of Instrumentation, (S1), pp. 768-770, 2006.

[3] Q Wang, “Research on Formation Method of Multi-robot, ” Research Institute of Management Science and Industrial Engineering, 2017: 6.

[4] G. Barthe, T. Rezk, A. Russo and A. Sabelfeld. “Security of multithreaded programs by compilation,” ACM Transactions on Information and System Security (TISSEC), 13(3), 2010.

[5] H Ren, “Behavior Control System Design of NAO Robot,” University of Jinan, 2016.

[6] N Wang, “Research on Robot Control Technology Based on Speech Recognition ,” Northeast Petroleum University，2011.

[7] T BALCH , R C ARKIN, “Behavior based formation control for multi-robot teams,” Transactions on Robotics and Automation, 14(6): pp. 926-939, 1998.

[8] W Zhang，Y Fu，Z Guan and H Wang, “Application of Windows Multithreading in Multi-robot Coordination Control,” Computer and Digital Engineering，(04): pp. 4-7, 2005.

[9] L Jia，L Wang，Z Xing and J Han, “3R Robot Control System Based on Multithreading Technology,” Microcomputer information，(17): pp. 243-245, 2017.

[10] E A Macdonald, “Multi-robot Assignment and Formation Control,” Georgia Institute of Technology, 2011: 76.

[11] A Russo and A Sabelfeld, “Securing interaction between threads and the scheduler in the presence of synchronization,” Journal of Logic and Algebraic Programming, 78(7), 2008.

[12] T Justin. Maris,Aaron W. Keen,T Ishihara,Ronald A. Olsson. “ A comparison of concurrent programming and cooperative multithreading under load balancing applications,” Concurrency and Computation: Practice and Experience, 16(4), 2004.

[13] D Wu, G Zeng, L Meng, W Zhou, L Li, “Gini Coefficient-based Task Allocation for Multi-robot Systems With Limited Energy Resources,” IEEE/CAA on Journal of Automatica Sinica, 5(01), PP. 155-168, 2018.

[14] H FANG, C SHANG and J CHEN, “An optimization-based shared control framework with applications in multi-robot systems,” Science China(Information Sciences), 61(01), PP. 261-263, 2018.

[15] C Lesire, G Infantes, T Gateau and M Barbier, “A distributed architecture for supervision of autonomous multi-robot missions, ” Autonomous Robots, Vol. 40 (7), 2016.

[16] G A. Kaminka, A Shapiro. Editorial: “ Annals of Mathematics and Artificial Intelligence special issue on multi-robot coverage, ” search, and exploration, Annals of Mathematics and Artificial Intelligence, Vol. 52 PP. (2-4), 2009.