A Prototype of Remote Robotic Hand Controlled by FPGA

Ludi Wang†1 Takeshi Ohkawa†1

Abstract:

Remote Robotic Hand is required to solve problems safely and efficiently in dangerous situations. In this report the design and implementation of an FPGA-based robotic hand with remote control is proposed. Coordinate information of operator's hand is collected by a motion capture device. The robotic hand will be controlled by PYNQ on which its FPGA part accelerates the coordinate calculation. A preliminary evaluation of the prototype system has been done in terms of processing and network communication latency.

Keywords: FPGA, Robot hand, motion capture, remote control

1. Introduction

IoT systems design is a hot application field which creates low-power and low-cost robots in Industry 4.0 [1]. Specifically, with the creation of Intelligent robots that can help us in life and work. Remote robotic hand [1] is expected to be a tool to replace human hands. For example, doctors can operate on patients with infectious diseases without touching them. Police can defuse the bomb without touching it. The safety of the users is ensured by introducing remote robotic hand. Using the remote robotic hand, anyone can repeat the operator's hand movements.

In this paper, we propose the design of a remote robotic hand controlled by FPGA that can recreate hand movements of coordinate information of operator's hand collected by a motion capture device. PYNQ [2] (ARM+FPGA) was chosen for accelerating coordinate calculation by FPGA. Because there is a directly connected communication bus between CPU and FPGA. Therefore, the communication is expected to be faster, and the control system of the robotic hand can be simple and efficient, which results in small processing latency of remote control.

2. Requirement

The remote robot hand is controlled by motion capturing of the operator's hand. Each finger's joints can be moved individually. A motor for the joint is controlled by pwm. The total delay of motion capturing, network communication, data processing and pwm output for the motor control is expected to be within 0.1s.

3. Proposal

We propose a robotic hand controlled with a combination of Leap Motion [3] and PYNQ. Data collected by Leap Motion is transmitted to PYNQ through internet. The whole system is composed of the following elements, while the main contributions of this research are (B) and (C).

†1 Tokai University

(A) Motion capture of operator's hand

The captured hand coordinates are used to precisely control the movement of the robotic hand.

(B) Communication for remote control

A cloud service "PubNub" [4] is used for data transfer. The Internet allows operator to control it from anywhere.

(C) FPGA acceleration of coordinate processing

An effort is made to improve motion accuracy and latency reduction. Speeds up the processing by FPGA.

(D) Robot hand with independent fingers

The manipulator consists of five independent fingers. All joints are controlled by their own motors. Each of the joints on the fingers can move independently. This means that every joint in the hand can move independently. The user can use the robot to fully recreate the hand movements.

4. Evaluation

4.1 Method

The development of software is performed in Python using the Leap Motion [3] API libraries and the LeapCSharp.NET4.0.dll.

First, Leap Motion captures the coordinates from operator's finger. Data will be transmitted over the Internet to PYNQ for data analysis. The PYNQ will acquire the data and perform a smoothing of the coordinate data. After that PYNQ will calculate the speed of the finger movement and the angle between the fingers.

The next step includes calculate the robotic finger's joint final position and required speed with the help of inverse kinematics transformation. Eventually, the motors control the robotic movements.

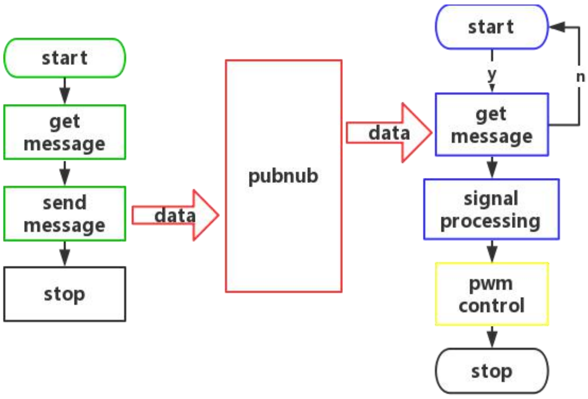

Fig. 1 The flow of the prototype system of a remote robotic hand

The server (left part of Fig.1) sends the name of the finger (middle finger) in the finger channel as the starting signal. The server sends the name of the bones on the bone channel as an identification signal. Identify the different bones.dir0 sends the x-coordinate of the bone.dir1 sends the y-coordinate of the bone.dir2 sends the z-coordinate of the bone.

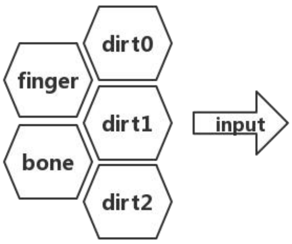

Fig. 2 Collecting and inputting data

On the client side (right part of Fig.1), “finger name” + “four bone names” + “corresponding coordinates” make up a group (*Fingerlist*). It starts averaging when data for five groups is received. Then, five groups of x-coordinates of a bone are processed by removing a maximum value, removing a minimum value, and averaging the rest. The final x value is obtained. After all a *Fingerlist* will end up with four average coordinate points.

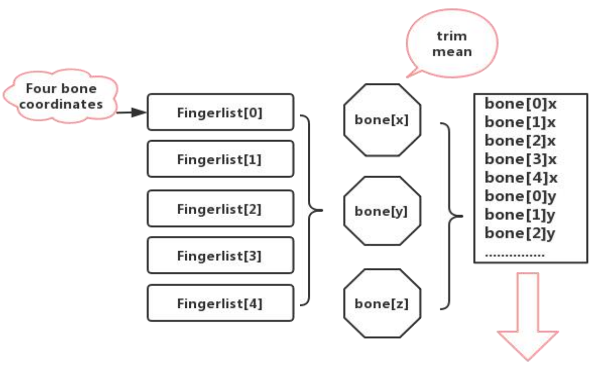

Fig 3. Data grouping and data processing

4.2 Comparison to traditional technologies

In contrast to software-based solutions with real-time operating systems, a deterministic behavior can be achieved using FPGAs. Complicated calculations can be carried out in an extremely short time. Therefore, it is much faster and more efficient to use FPGA to process the acquired hand motion information. The FPGA has the advantage of low power consumption. This means that the robot can work for a long time even with limited power.

4.3 Preliminary Experiment

Raspberry Pi and PYNQ-Z2 board have similar power consumption,computational performance, and GPIO configuration. It can be used as a control test. Compare the computational latency of the two. The measured processing and communication time is shown in Table 1. When the server sends data, it records the start_time of communication, and it also records the end_time after receiving the data. After receiving the data, the start time of data processing is recorded. After calculating the coordinates and angles of the five fingers, the end time of processing is recorded. This way the time of data processing and communication can be calculated.

Table 1 Measured processing and communication time

Raspberry Pi	Elapsed Time (s)
Network communication	0.003
Data processing	1.202

5. Summary

In order to investigate the processing latency of a remote robotic hand system, a prototype system has been developed. The measured processing time is much longer than the network delay and should be accelerated. In the future work, the FPGA part of PYNQ will be utilized to speed up data processing of the coordinate calculation.

Reference

[1] C. Paparizos, N. Tsafas and M. Birbas, "Design and Implementation of an APSoC-Based Robotic System with Motion Tracking Teleoperation," 2019 8th International Conference on Modern Circuits and Systems Technologies (MOCAST), Thessaloniki, Greece, 2019, pp. 1-4, doi: 10.1109/MOCAST.2019.8742037.

[2] “What is pynq” [Online] Available: http://www.pynq.io/home.html

[3] “Leap Motion Controller” [Online] Available: https://developer.leapmotion.com/sdk-leap-motion-controller/

[4] “PubNub” [Online] Available: https://www.pubnub.com/