Design, fabrication, and programming of 5-fingered underactuated robotic hand for use with ROS

Charles DeLorey

Department of Computer Science, Tufts University

May 10, 2020

Abstract

This article presents a 3D printed 5-fingered robotic end effector and associated codebase for use in research and academic settings. Fabricated from plastic and powered using an Arduino Uno microcontroller, the system is integrated with the Robot Operating System (ROS). The hand acts as a ROS node, and takes commands that other ROS programs can send to set configurations of the 5 fingers. The results presented in this paper demonstrate the feasibility and accessibility of such a project for other microcontroller-based tasks required for a ROS-based system.

Keywords

Robot, manipulator, ROS, rosserial, 3D printing, anthropomorphic, Arduino.

1 Introduction

This paper presents a simple and low-cost 5-fingered robotic hand that uses the Robot Operating System (ROS) as a control system. The end effector designed and implemented operates on a open-loop control system, which means there is no feedback provided (i.e. no publisher of the current joint states from the hand) provided to the ROS node issuing desired joint angles.

An end effector is a device that is attached at the end of a robot arm or other limb that is used for interacting or manipulating the environment. Robot end effectors are used in a variety of settings including manufacturing and research. These end effectors come in a wide variety of shapes, sizes, configurations, and purposes. When designing tools with humans in mind it is important to take the design of our own innate end effectors into consideration. For the purposes of this paper, the term "robot[ic] hand" is used interchangeably with "end effector".

The human hand is an impressive and complicated system, made up of many subsystems; from the muscular system providing motion to the nervous system, which provides control signals to organs/organ systems. Though robotic hands made for prosthetics need to meet certain weight, form factor, and usability metrics, such devices made for research can be more flexible in their specifications. 3D printing, or additive manufacturing, has greatly increased the rate at which prototype parts can be designed, fabricated, and evaluated. In the field of robotics, 3D printing has made creating parts incredibly easy, and access to resources from websites such as Thingiverse [1] mean users can share designs incredibly efficiently. There are many impressive anthropomorphic robotic hands available for manufacturing or research, however the cost alone is often a barring factor in using said technologies [2]. Therefore, this project works to present a more cost-friendly yet robust solution to the problem of useful end effectors.

The order of sections are arranged as follows. Related Work introduces research using 3D printing, underactuation, and the rosserial package. Technical Approach documents the design and fabrication of the hand, in tandem with the development of the Arduino and ROS code. Finally, in the Conclusions and Future Work sections we summarize the work completed, as well as plans for future iterations on this project.

[1] https://www.thingiverse.com/
[2] www.shadowrobot.com/products/dexterous-hand/

2 Related Work

3D printing has become a widespread tool for rapid prototyping of designs. As an incredibly versatile technology, it has been used in many different applications, from construction to biotechnology. Immense 3D printers extruding cement have been created for large-scale construction of architectural components [1]. In the biotech field, 3D printing is being used to produce 3-dimensional cell cultures for more precise scaffolding of structures in tissue engineering [2]. The customizable nature of 3D printing and Computer Aided Design (CAD) software enables user-driven customizations to better suit their specific needs. There have been projects investigating the use of 3D printing in manufacturing robotic hands. Research in using flexible materials to better mimic the natural movements and dynamics of the hand shows an effective use of a biologically inspired design implemented with 3D printing technology [3]. 3D printing also allows for modularity which is a valuable feature in robotics, especially open-source robotics [4].

Another important aspect of end effector design is the method(s) by which a device is actuated. A common mechanism for end effector actuation is underactuation [5] [6]. This means the number of actuators (motors, etc) is less than the number of degrees of freedom (DOFs) in the system. The opposite of this are fully actuated systems, wherein they have exactly the number (or more) of actuators for each DOF. Fully actuated systems lend themselves to precise motions, but appear too unnatural and often require more energy than necessary for a given problem [7]. The proposed method of actuation reduces the number of motors in the system, which for the scale of this project is ideal. Underactuated robotic hands are able to approximate fairly well the range of motion of a human hand [3].

Trying to mimic the properties of human anatomy and the behavior of the materials which make up our hands is a difficult task. There is research into using flexible or otherwise compliant materials in constructing robots in the effort to give them better gripping capabilities, as well as make them safer around humans [3]. Creating functional models of the human body advances understanding of biological mechanisms, such as bipedal locomotion, as well as provide useful data to be applied in the field of robotics [8].

An invaluable reference for this project has been the InMoov robotic hand, as part of a project to create an open-source life-size robot[9]. The hand uses a 5 servos located in the forearm of the robot to actuate the fingers. Open-source robotics also contribute to the field of prosthetics, as the open-source initiative OpenBionics has GitHub repositories featuring ROS code for their robot hands [10].

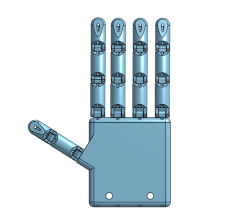
Hardware is an important factor in robotics, as there is a delicate balance between hardware size and efficiency versus the computational requirements of the problem. Microcontrollers such as Arduino, Raspberry Pi, and XBEE benefit from ROS support of serial-based devices and microcontrollers using the "rosserial" package. In [11], the authors use Arduino for its ease of use, plethora of tutorials and libraries, and inexpensive components. Ibáñez et al. [12] use an Arduino Uno for low-level control of direct current (DC) motors. This enables them to precisely control their robot, while an onboard computer handles navigation and planning using Simultaneous Localization And Mapping (SLAM), an important problem in mobile robotics. In this paper we use the "rosserial_arduino" library, written for Arduino (in the C++ language). Arduino and the rosserial_arduino library were chosen for this project for its easy to use interface, availability of ROS libraries/integration, and previous familiarity with the platform.

3 Technical Approach

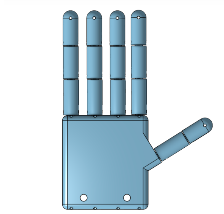
The 5-fingered anthropomorphic hand design provides a close approximation to the geometry and grip configuration of a human hand. As discussed in the Related Work section, creating approximations of the human hand provide insight into human-oriented design, as well as advance the field of anthropomorphic robotics. The hand features 5 fingers, 4 of which are made up of 3 segments. These are akin to the distal, intermediate, and proximal phalanges, or bones that make up fingers of a human hand, going from the tip of the finger towards the palm [13]. The thumb, unlike a normal human hand, features only 2 finger segments separate from the palm, to better reflect the natural articulation of the thumb in relation to the palm. The Arduino code works as a ROS subscriber, receiving joint messages and configuring the hand. The ROS code acts as the handler for the serial connection, enabling other nodes in the network to send messages to the hand. The following sections detail the physical hand design, the Arduino code, and the ROS code.

3.1 CAD & Fabrication

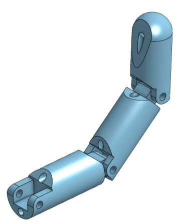
An approximation of the palm and fingers of a human hand were modeled from standard shapes (cylinders, rectangular prisms) in OnShape [3], a browser-based full-cloud CAD software. The design of the hand is based upon the geometry of a human hand, and the actuation is informed by the musculature. There are two types of muscle groups of the hand which perform finger flexion and extension, as well as circumduction (rotation) of the wrist. Intrinsic muscles are those located within the hand, and are the muscles responsible for fine motor control. Extrinsic are those that act on the hand but are located elsewhere, namely in the forearm, and are what allow for gross motor movements and strength. The intrinsic muscles are the ones of interest in this paper, and as such are simulated by the hand designed. Anatomically speaking, many of the extrinsic muscles are involved in finger extension, and in a sense the "extrinsic" nature of the muscles is simulated by the design of the servos actuating the fingers [13]. The movement of the fingers is achieved using two strings, each threaded through the length of the finger and attached at the fingertip. One string is threaded through on the palm-side (palmar), and the other on the back-of-the-hand side (dorsal). Each pair of strings for a finger are connected to the same servo motor on either end of a servo horn (bar-shaped plastic which is meshed with the servo gear, enabling rotation). When the servo rotates one direction, it pulls one of the strings, achieving flexion or extension of the finger. The joints of the fingers (3 per finger) are articulated with M2 16 millimeter bolts, and held in place with M2 size nuts.



(a) Palmar view



(b) Dorsal view



(c) Flexed finger

Figure 1: Views of hand CAD model and finger assembly

The palm is modeled as one piece (see Figure 2), and contains both holes for pivot points of the fingers (a.). Also, there are 8 holes that span the length of the palm, running proximally (towards the midline or "center" of the body) (b.). These act as guides to channel the strings to the servos, helping manage the strings, as well as discretize the movement space needed by each servo controlling its respective finger.

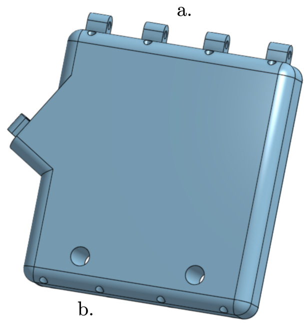
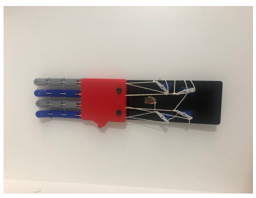
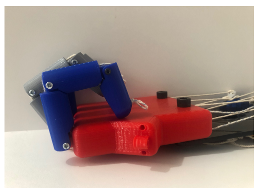


Figure 2: Palm model, labeled

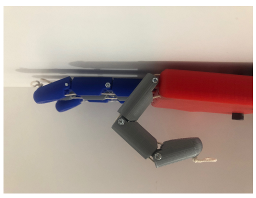
Figure 3 shows the robot hand from various angles. The servos are affixed to a plank of acrylic, which is attached with two nuts and bolts to the palm of the hand. Further design iteration will yield a more presentable arrangement for this part of the hand. In Figure 3.(d) you can see the servo horns, to which the strings are attached. Figure 3.(c) shows the sideways view of the hand, and the two strings threaded through to meet at the fingertip.



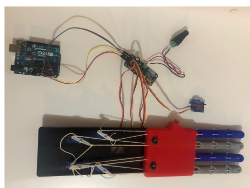
(a) Top view



(b) Left view



(c) Right view



(d) Top view with electronics

Figure 3: Views of hand

The electronic design of this project requires an external power supply, as well as a USB connection to the PC running the ROS network. Figure 4 shows the connection layout of the various systems. The Arduino receives power and

[3]https://www.onshape.com/

serial connection to the PC through a USB to mini-USB cable. The servos, however, are provided their own power from the motor driver, which in turn gets the power form an AC/DC power adapter. The external power source is necessary, as powering the servos from the Arduino USB connection would cause brownouts or other electrical problems for the system. At present, the robot hand does not have power and data connections that easily integrate into a larger robotic platform, such as a robotic arm.

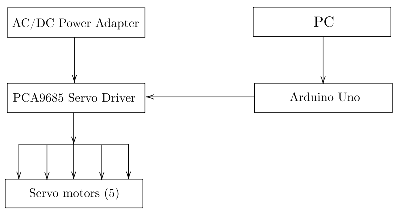


Figure 4: Electronic diagram

3.2 Arduino Code

To communicate with ROS on the PC (host side) the Arduino uses the ros_lib library, which contains message definitions as well as necessary code for creating and using publishers/subscribers. It uses the serial port connecting the Arduino to the PC (which is used for uploading a compiled executable program to the Arduino). The code compiled and uploaded to the Arduino creates a subscriber to the "servo" topic, awaits a ROS message containing joint positions, and then sets the servos of each finger accordingly. The Arduino effectively becomes a ROS node, able to publish and subscribe to topics over the serial connection. To control the servos, the HCPCA9685 library provides an easy interface for sending certain positions to a given servo [4]. The Arduino is connected to a PCA9685 servo driver board (see Figure 5), enabling each servo to receive 5v of power and be addressed individually with the inter-integrated circuit (I2C) protocol.

The Adafruit PCA9685 library, though designed to support their own product, an identical PCA9685 board, was deemed rather finicky to use; the HCPCA9685 library allows for more streamlined interaction with the hardware [6].

[4]https://forum.hobbycomponents.com

[5]https://learn.adafruit.com/16-channel-pwm-servo-driver

[6]https://github.com/adafruit/Adafruit-PWM-Servo-Driver-Library

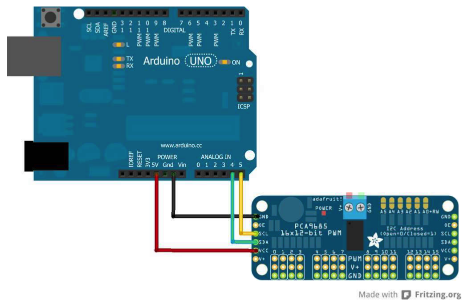


Figure 5: Arduino to PCA9685 wiring [5]

This control system is an open-loop control system, as the robot hand node does not publish its current joint state, it only receives joint commands. This feature is to be developed in future iterations of the device.

3.3 ROS Code

This project was developed on ROS Kinetic in an Ubuntu distribution 16.04 machine, and all packages used in this paper are compatible with this version of ROS. The PC (host-side) runs rosserial, as well as any other ROS nodes used in a given overall network. Rosserial is the protocol by which ROS messages are serialized and sent over a network socket or serial port, the latter being what Arduino uses in this system. Specifically, the host-side uses the package rosserial_python to create a serial node that facilitates the connection between the ROS network and the serial device. It connects to the serial port of the Arduino, usually /dev/ttyACM0. The serial node handles all the publishers and subscribers for the device, which in this case is the Arduino. To configure the hand from the host-side is as straightforward as publishing a message to a topic in a typical ROS node.

4 Conclusion

This paper documents the design, construction, and establishment of a code base for a 5-fingered robotic hand powered with an Arduino microcontroller. Related research with 3D printing, underactuated robotics, and prosthetic hands inform the design decisions of this project. Fabricated from 3D-printed plastic, the fingers are underactuated with two strings attached to servos. The Arduino program of the hand acts as a ROS node, taking finger joint commands. This project presents as a functional implementation for the rosserial series of packages, enabling users of ROS to develop their own systems using this design and codebase. Limitations of this project, due to time and resource constraints, lie with the restricted range of motion of the fingers, and the unidirectionality of the ROS messages that can be sent.

5 Future Work

The primary focus of any further work on this project is refining the codebase to have a standalone, hardware-agnostic system. For example, if the end effector being constructed had only 2 fingers, or used a different form of actuation. Furthermore, adding a publisher feature so other ROS nodes can interrogate the hand and subscribe to its current joint positions. It is possible that the Arduino will be shown not able to function as a publisher and subscriber simultaneously, at which point more research will be conducted into other microcontrollers. Furthermore, the hand CAD model warrants additional iterative design to be improved in both biological accuracy and mechanical efficiency. Due to the limited scope and erratic timeframe of this project, thorough experimentation was not achieved. Future developments with this project will also include an evaluation of the hand in grasping and other tasks, in particular when integrated with a robotic arm (such as the UR5), as well as an efficiency comparison to other similar systems.

6 Acknowledgements

I wish to thank my major advisor, directed study advisor, and PI, Professor Jivko Sinapov, for his support of this project as well as welcoming me into his robotics lab and research opportunities. Without these experiences I probably would not have undertaken this project. Additionally, I am particularly grateful for the assistance and support given by fellow Tufts computer science students, as sounding boards and as supportive individuals.

The GitHub repository containing parts information, CAD files, and code can be found here: https://github.com/cdelor02/robothand

References

[1] Buswell RA, de Silva] WRL, Jones SZ, Dirrenberger J. 3D printing using concrete extrusion: A roadmap for research. Cement and Concrete Research. 2018;112:37 – 49. SI : Digital concrete 2018. Available from: http://www.sciencedirect.com/science/article/pii/S0008884617311924.

[2] Zhu W, Ma X, Gou M, Mei D, Zhang K, Chen S. 3D printing of functional biomaterials for tissue engineering. Current Opinion in Biotechnology. 2016;40:103 – 112. Tissue, cell pathway engineering (2016). Available from: http://www.sciencedirect.com/science/article/pii/S0958166916300702.

[3] Tian L, Liu J, Thalmann NM, Thalmann D, Zheng J. Design of a Flexible Articulated Robotic Hand for a Humanoid Robot. In: 2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids); 2019. p. 572–577.

[4] Ma RR, Odhner LU, Dollar AM. A modular, open-source 3D printed underactuated hand. In: 2013 IEEE International Conference on Robotics and Automation; 2013. p. 2737–2743.

[5] Santina CD, Piazza C, Grioli G, Catalano MG, Bicchi A. Toward Dexterous Manipulation With Augmented Adaptive Synergies: The Pisa/IIT SoftHand 2. IEEE Transactions on Robotics. 2018;34(5):1141–1156.

[6] Gosselin C, Pelletier F, Laliberte T. An anthropomorphic underactuated robotic hand with 15 dofs and a single actuator. In: 2008 IEEE International Conference on Robotics and Automation; 2008. p. 749–754.

[7] Tedrake R. Underactuated Robotics: Algorithms for Walking, Running, Swimming, Flying, and Manipulation (Course Notes for MIT 6.832). MIT;. Available from: http://underactuated.mit.edu/.

[8] Powell MJ, Zhao H, Ames AD. Motion primitives for human-inspired bipedal robotic locomotion: walking and stair climbing. In: 2012 IEEE International Conference on Robotics and Automation; 2012. p. 543–549.

[9] Langevin G. InMoov-Open Source 3D printed life-size robot.; 2014. Available from: http://inmoov.fr/.

[10] Zisimatos AG, Liarokapis MV, Mavrogiannis CI, Kyriakopoulos KJ. Open-source, affordable, modular, light-weight, underactuated robot hands. In: IEEE/RSJ International Conference on Intelligent Robots and Systems; 2014. p. 3207–3212.

[11] VanHuy T, Minh DT, Kien NP, Vũ TA. Simple Robotic Hand in Motion Using Arduino Controlled Servos; 2017. .

[12] Ibáñez AL, Qiu R, Li D. A Simple, Cost-Effective and Practical Implementation of SLAM Using ROS and Arduino. In: 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData); 2017. p. 835–840.

[13] Saladin KS. Anatomy physiology: the unity of form and function. McGraw-Hill Higher Education; 2010.