



UNIVERSITÀ
DI SIENA
1240

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE E SCIENZE

MATEMATICHE

Computer and Automation Engineering

ROBOTICS AND AUTOMATION

Closed loop approach to human-robot handshake

Supervisor:

Chiarissimo D. Prattichizzo

Co-supervisor:

? **M. Malvezzi**

? **E. Knoop**

Student:

Francesco Vigni

A.A. 2017-2018

Contents

1	The state of the Art	6
1.1	The Idea	7
2	Hardware setup	8
2.1	The Pisa/IIT SoftHand	9
2.2	The Sensors	10
2.2.1	Calibration of FSRs	11
3	Software setup	15
3.1	Ros	15
3.2	Nodes	16
3.2.1	Pisa/IIT SoftHand node	16
3.2.2	FSRs node	17
3.2.3	Auxiliary nodes	17
4	Open Loop Experiments	18
4.1	Step input	20
4.2	Pseudorandom input	20
5	Closed Loop controllers	21
5.1	Empirical Proportional controller	21
5.2	Human-calibrated controller	21

CONTENTS

3

6 Results

22

Abstract

The following work is focusing in the Human-Robot interaction, specifically in the dynamics of a handshake. The handshake event between human beings is a well known task, it is able to transmit participants feelings as a mixture of physical features like: strength of the handshake, velocity approach, duration of the handshake, oscillation frequency and amplitude of the arm. The target of this work is to set up an environment in order to test different closed loop controllers for the handshake event. The environment is built using Robotic Operating System (ROS), which is managing messages among: a Pisa/IIT SoftHand, four FSR 400 sensors managed by an Arduino Uno and the auxiliary nodes. The work is splitted in two main experimental parts: calibration and tests. The calibration part is built as an open loop system in which we're interested in understanding the participants grasping force with respect to Pisa/IIT SoftHand's force. The hypothesis is to model the human response to the robot grasp as a dynamical system. Different dynamical models can explain the data retrieved and are tested with closed loop controllers.

Introduction

The handshake event, commonly used, is a natural human interaction and is extensively used worldwide in events like: greetings, introduction routine between human beings and agreements. The scientific approach to handshake between a human and a robot, therefore, must intrinsically deal with low levels human interactions and some assumption must be done in order to focus on the task. The handshake event can be divided in two parts: the approaching and the handshaking. This work is focused on the haptic sense involved during the handshake, knowing that the approaching is mainly executed using non haptic senses(f.i. vision).

A parameter involved in the handshake is the consensus; it allow the human to evaluate a handshake mixing aspects like: duration of the event, dynamics, force exchanged. An important part of this work is to test different controllers with the purpose of evaluating the consensus using closed loop controllers.

Chapter 1

The state of the Art

Develop a robot capable of performing a smooth human-like handshake is still a highly interested topic in the scientific literature. A natural handshake between two humans is a very complex task to replicate, this work just focuses on the interaction force between an artificial hand and a human hand. The consensus is a complex task to encode inside a robot, participants will easily distinguish the event with respect to another human or to a robot. A human will take into consideration the skin feedbacks like: the temperature, the humidity and the softness. These are some characteristics that are still not embedded into the hardware available in the market. The aspect taken in consideration in this work is the grasping force exchanged in the handshake. Robots, nowadays, are highly involved in industries where mostly they have to execute repetitive tasks. These kind of robots, when have to grasp objects, commonly uses multi purposes grippers [2], therefore, an accurate choice has been done on the hardware to use in this work.

[7] [8] [6]

1.1 The Idea

The idea is to create a closed loop controller for the human-robot handshake event, using Pisa/IIT SoftHand produced for research purposes at Università degli studi di Pisa and Istituto Italiano di Tecnologia (IIT) and upgrading it with four independent FSR sensors which uses an Arduino uno in order to communicate the data. The FSR sensors are located on the robotic hand so there are no wearing device on the human hand during the execution of the experiments. This choice leads the work to be focused on the theoretical part of the handshake event, and potentially reach robust results. We are focusing on a general human-robot handshake, knowing that the interaction can vary with participants, f.i. the participant's hand size is affecting the firsts contact points or nominal strength to apply in the handshake can be affected by prior expectations. It is more meaningful then, to study individual differences once the generic case has been studied. The chosen robotic hand (Pisa/IIT SoftHand) has 19 degrees of freedom and its main characteristic a single dc motor that is pulling a tendon which is embedded in each finger. This physical approach results in an under actuated robotic hand which can be controlled only by the dc motor and can easily adapt to different configurations without modifying the reference position.

Chapter 2

Hardware setup

The hardware must be physically merged in order to reach the goal of a closed loop controller for a handshake. From early approaches as [5], an estimation of the human palm has been used. The *sensorized palm* idea was to approximate the human palm which is the part of the hand mostly interested during a handshake. The figure 2 shows the position in which the sensors are placed, in particular the n. 1, 2, 3 are placed according studies in [5], the sensor n. 4 is placed in order to increase the weight of human-thumb force applied during the grasp.



Figure 2.1: Pisa/IIT SoftHand with FSR sensors for handshake

2.1 The Pisa/IIT SoftHand

The Pisa/IIT SoftHand is a simple, robust and effective hand designed for grasping and soft manipulation presented in [1], the hardware is provided with a controller developed by the same group which implements a proportional controller Fig. 2.3 on the motor position. This enables the researchers to control the Pisa/IIT SoftHand with a reference position, without dealing with the current control of the motor. A tick scale is provided with the device in order to set a range of possible closure positions, more formally, let's define $q \in \mathbb{N}$ as the closure position of the hand such that $\max(q) = 19000$ (Pisa/IIT SoftHand fully closed) and $\min(q) = 0$ (Pisa/IIT SoftHand fully opened).

The proportional coefficient can be set up as preferred since its encoded in ROS

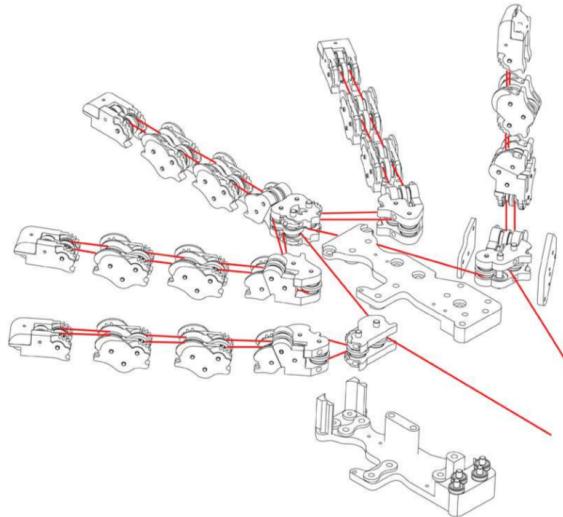


Figure 2.2: Exploited view of the modules of Pisa/IIT SoftHand

as a *rosparam*, this parameter is thought to range between 0 and 1.0. Setting the parameter to 1.0 is minimizing the error value $e(t)$ between the setpoint $r(t)$ and the output $y(t)$. The successful idea in the design of Pisa/IIT SoftHand can be found in the flexibility of the joints and the wide range of usage. The reference position $r(t)$ has no 1:1 correspondence with the single physical position of each

finger. Having a single motor to control makes the robotic hand really easy to control but introduce uncertainty on the position of each finger. A tendon is running through all the fingers and is pulled by the internal dc motor, therefore, the only available information is the overall tick-position of the Pisa/IIT SoftHand. This device has an internal value returning to the system the real tick position, this value is compared with the referenced one in the controller. The real tick position is a value that must be calibrated manually using administrative tools provided by the manufacturers. The calibration is manual which means that the Pisa/IIT SoftHand is manipulated to be into a fully open position and the program save that position as the zero tick position.

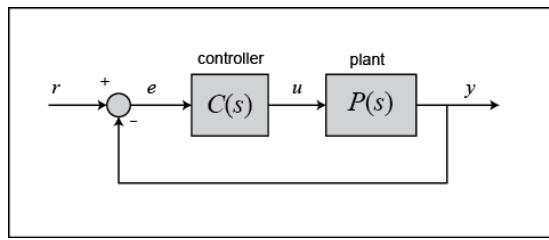


Figure 2.3: Block Diagram Proportional controller in a feedback loop

2.2 The Sensors

The core of the closed loop control is to have a feedback in the whole system which is proportional to the force applied during the handshake from the human.

The sensors used in this work are Force Sensitive Resistors Fig. 2.4 measuring the force applied from the human to the Pisa/IIT SoftHand, in order to decouple this force from the one applied from the Pisa/IIT SoftHand to the human hand, [5] has been used and the physical position of the FSRs has been setted up accordingly. FSR sensors are devices that allow to measure static and/or dynamic forces applied on the sensing area, through the variation of its electric resistance. The main advantage of these devices is the low cost per-unit, little space required

for installation (thickness under 1.25mm) and the force sensitivity range up to 20N.

As robust polymer thick film devices, the FSRs, exhibit a decrease in electric resistance with increase in force applied to the surface. By theory is considered that when a force is applied the resistance changes approximately linear in a logarithmic plot [4]. A simple force to voltage conversion is physically implemented as suggested the manufacturer, in fig. 2.5 is shown a snippet of the above cited data sheet. For this work RM is fixed to $3.3k\Omega$. These mentioned sensors are the

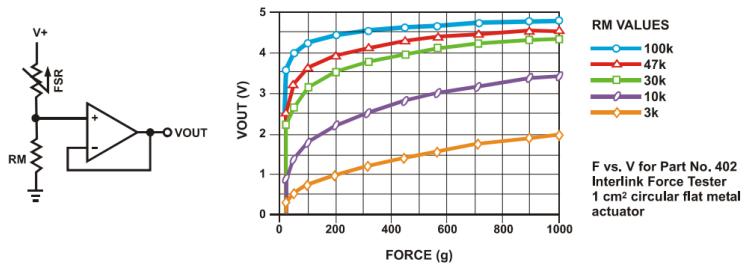


Figure 2.5: FSR Datasheet snippet

more natural choice for handshake experiments since their thickness keeps the size of the Pisa/IIT SoftHand reasonable for human-robot handshake. Ideally using more sensors allow to get more relevant data but the surface available on the Pisa/IIT SoftHand is limited. The choice in the number of FSR sensors comes from the trade off between using lots of FSR sensors but with a smaller area and using a smaller amount of sensors but with higher area. The first configuration does not ensure the contact among experiments with different participants and the second configuration leads to physical bending of the sensors and influences the consistency of the readings.

2.2.1 Calibration of FSRs

The calibration procedure of a sensor is a really important task, since it allows to compare experiments and to provide consistent results. The first voltage-to-force relation for the FSRs comes from a manufacturer sketch which is returning force with measures standard of grams. As shown in [3], load cells can be used as 'ground truth' to calibrate force sensitive resistors to provide informations in Newtons. Using a *sensorized palm* developed in [5], sketched in Fig. 2.6 which embeds a load cell and placing the FSR sensors accordingly with the main direction of the force on this *sensorized palm*; values from FSRs and the load cell are compared.

Mathematical regression tools have been used in order to find a model that explain the values from the sensors compared to the force of the load cell. Although an exact calibration of FSR sensors is not the target of this work, a first order model has been fitted to the data in the force-range of interest of this work. The number of FSR sensors involved in the calibration experiment is two, instead the amount of FSR sensors chosen for the human-robot handshake is four. The configuration for the calibration experiment, with the *sensorized palm* and two FRS sensors is shown in Fig. 2.7. In order not to add complexity to this task, is assumed that using two FSR are sufficient to fit a model to the FSR sensors involved in the handshake environment.

The experiment is executed with a participant applying different forces on the above mentioned device, and recording the output from the FSR sensors and from the load cell of the *sensorized palm*. The plot in Fig. 2.8, shows the data obtain from this experiment. A toolbox for data fitting from Matlab has been used in order to find the best linear fitting equation, according to the Least Mean Square Error to the raw data, which analytically is:

$$y = 135.8x + 1336 \quad (2.1)$$

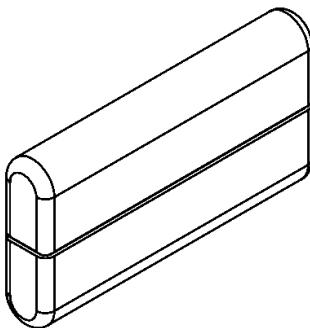


Figure 2.6: Sensorized palm

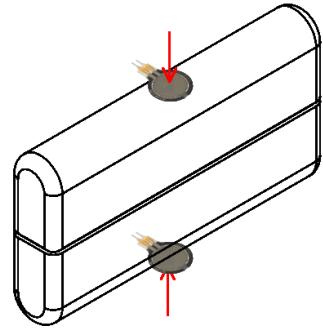


Figure 2.7: Sensorized palm with FSR sensors

where y is the value taken as the sum of the FSRs sensors, in this experiment two FSRs has been used. The variable x is the value from the *sensorized palm*, returning Newtons, used to find the best first order model to map the FSRs values.

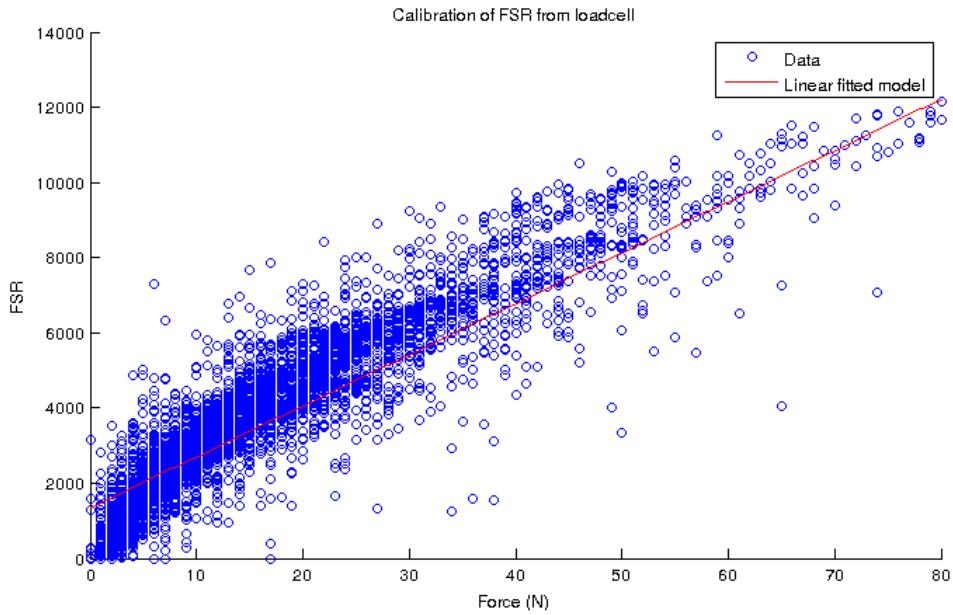


Figure 2.8: FSRs vs. Load Cell data

Obviously, more advanced models can be fitted in order to have FSR sensors returning Newtons with a higher precision, but for the specific task required in this work is not necessary to increase the degree of the model. The handshake

event is involving grasping forces in the range of the above-mentioned calibration so the force-plots in the next chapters will be intended to be Newton scaled.



Figure 2.4: FSR 13mm

Chapter 3

Software setup

The described experiments can be implemented using a software capable of exchanging informations between robots, without the interaction of a human. A tool named Robot Operative System has been chosen in order to manage the informations between the devices involved in these experiments.

3.1 Ros

The Robot Operative System (ROS) is a set of frameworks and libraries useful for robot software development. The logic of this software is really intuitive, it lets the developers to represent a device as node inside a graph. The most important node in this graph is the *Master*, which is managing all the messages in the system. Each node in order to send or receive messages to/from other nodes must communicate this intention to the *Master* node, which is processing the request and forwarding the right informations.

The ROS graph is done using a network of process called ROS nodes. The main concepts in the ROS graph are ROS Nodes, Master, Parameter server, Messages, Topics, Services, and Bags. Each concept in the graph is contributed to this graph in different ways. Each message in ROS is transported using named buses

called topics. When a node sends a message through a topic, then we can say the node is publishing a topic. When a node receives a message through a topic, then we can say that the node is subscribing to a topic but the production of information and consumption of it are decoupled. Each topic has a unique name, and any node can access this topic and send data through it as long as they have the right message type. The type of a message can be chosen among the standard primitive types (integer, floating point, Boolean, and so on) or we can define custom message types.

3.2 Nodes

The main aim of ROS nodes is to build simple processes, this makes debug easier and simplify the structure of a project. Each ROS node is written using ROS client libraries such as roscpp and rospy, in short a ROS node can be written as a C++ source file or as a python executable program.

3.2.1 Pisa/IIT SoftHand node

Pisa/ITT SoftHand is provided with a variety of ROS packages, in particular, ROS node *qb_force_interface* is the node managing the proportional controller on the position of the Pisa/IIT SoftHand. This node is publishing a topic named: /qb_class/hand_measurement which embeds a custom field structure named: qb_interface/handPos. Custom field structure of messages are useful in order to send different message type over the same channel (topic).

3.2.2 FSRs node

The FSRs are connected to a bare PCB, each one following the circuit diagram in Fig. 2.5; This allows the source code flashed on the Arduino to: read the voltage difference at FSRs terminals, apply the linear model for force measurements and publish the information to ROS. A specific protocol called *rosserial* is used in order to implement a ROS node with Arduino Uno.

3.2.3 Auxiliary nodes

The hardware nodes of the project are now up and running, but in order to manage the informations and to eventually save files during the execution of the experiments, auxiliary nodes are created. The auxiliary nodes are written in C++ or Python and the choice is strictly related to the compatibility with specific library (f.i. *SMACH* State MACHine is a library used to implement state machines, currently only available in Python). Several auxiliary nodes have been created in order to manage:

1. FSR calibration experiments
2. Saving data during experiments
3. Open loop experiments
4. Closed loop experiments

Again, ROS provide the concept of node which is extremely useful for splitting work in smaller and more manageable units.

Chapter 4

Open Loop Experiments

The open loop experiments aim to verify the hypothesis that the human response to a robot handshake can be modelled as a dynamical model. In these experiments human participants are intended to be a natural calibration system for the robot grasping force F_r . Participants are asked to apply the force that they are perceiving during the experiments. The force is exchanged during a handshake only after the reference position has reached the first contact point q_0 , therefore for values of $q < q_0$ no force will be exchanged in the handshake.

$$F_r(q) = \begin{cases} k_r(q - q_0) & \text{for } q - q_0 \geq 0 \\ 0 & \text{for } q - q_0 < 0 \end{cases} \quad (4.1)$$

Where F_r is the force during the handshake applied by the robot on the participant's hand, q is the reference position sent to the device. During these experiments each reference position of the Pisa/IIT SoftHand is held for 3 seconds, the frequency rate is set to 100Hz. A file standard has been created in order to compare different experiments. The file is a '.csv' file with columns [FSR1, FSR2, FSR3, FSR4, Current, Real Position, Reference Position]. All the

plots below are obtained as processing files with the previous structure. Each experiment starts with reference position set to 0 and finish with the reference position set to 0. The experiments are in open loop so in order to avoid injuries an emergency function has been created, if the participant starts feeling pain the key 'x' on the keyboard must be pressed. The Pisa/IIT SoftHand will take as reference position 0 (fully open) and the whole program will be stopped. In case of emergency the file will be saved and the last data can be used to understand the configuration just before the trigger event.

The experiments are done with the Pisa/IIT SoftHand in a horizontal position (palm facing down) Fig. 4.1. In this way the weight of the Pisa/IIT SoftHand will not affect the FSR readings.

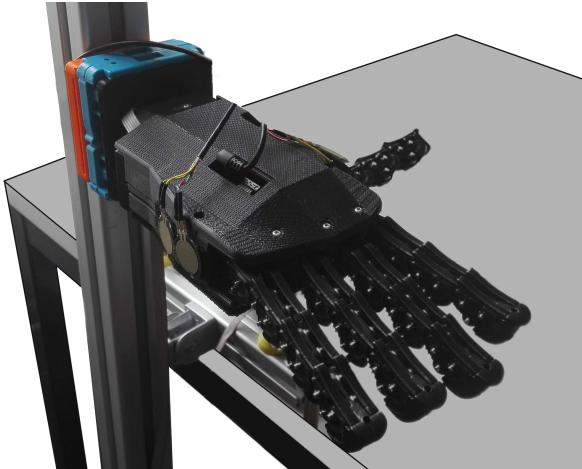


Figure 4.1: Palm facing down environment

4.1 Step input

The simplest signal that can be sent to the Pisa/IIT SoftHand is a step signal on the reference position, in this way the participant's response can be evaluated. The step signal in this experiment is formally a shifted and scaled step signal, the transformation parameters have been chosen in order to start from a position without physical contact ($k_0 = 8000$) to a position where empirical experiments have shown a consistent contact ($k_1 = 15000$). The first three seconds of the experiment k_0 is sent, then k_1 is sent so that the whole experiment lasts 6 seconds.

$$Q_r(t) = \begin{cases} k_0 & \text{for } t < t_0 \\ k_1 & \text{for } t \geq t_0 \end{cases} \quad (4.2)$$

A correlation between the reference position of the Pisa/IIT SoftHand and the values recorded from the FSRs is expected. The participants are naturally applying a force (F_h) which is supposed to be proportional to the one applied from the robot to their hand during the handshake. The same experiment has been executed with multiple participants, in order to increase the amount of data available for the model estimation. The values of reference position in this experiment are not changing during multiple trials, therefore the participants were able to predict that at $t = t_0$ a higher reference signal was sent. This behaviour has been considered not consistent to apply in post processing a model fitting of the data.

4.2 Pseudorandom input

Chapter 5

Closed Loop controllers

5.1 Empirical Proportional controller

5.2 Human-calibrated controller

Chapter 6

Results

Conclusion

This project applies learning [?] techniques to MNIST handwritten dataset. As we can see in the previous confusion matrix the accuracy of the final work is 97.6%. The overall idea is to train *autoenc1*, *autoenc2* and *softmax1* once per time and to crop the nets in order to have coherents dimension between network interconnections. At the end of [?]this process we stack all the partial neural network together and the deep neural network come to life.

The satisfaction behind this project can be experimented by running the file "MNIST_drawsim.m" which is a matlab function that allows the user to draw a digit and returns the correct digit value 97,6 times over 100.

Bibliography

- [1] M. G. Catalano, G. Grioli, A. Serio, E. Farnioli, C. Piazza, and A. Bicchi. Adaptive synergies for a humanoid robot hand. In *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, pages 7–14, Nov 2012.
- [2] M. Choi, D. Lee, H. Park, Y. Kim, G. Jang, Y. Shin, J. Park, M. Baeg, and J. Bae. Development of multi-purpose universal gripper. In *2017 56th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, pages 1421–1424, Sept 2017.
- [3] J. A. Flórez and A. Velásquez. Calibration of force sensing resistors (fsr) for static and dynamic applications. In *2010 IEEE ANDESCON*, pages 1–6, Sept 2010.
- [4] Interlink Electronics Inc. Pn: 34-00015 model 402 short tail. In *FSR 402 Data Sheet*.
- [5] E. Knoop, M. Bächer, V. Wall, R. Deimel, O. Brock, and P. Beardsley. Handshakiness: Benchmarking for human-robot hand interactions. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4982–4989, Sept 2017.

- [6] D. Papageorgiou and Z. Doulgeri. A kinematic controller for human-robot handshaking using internal motion adaptation. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5622–5627, May 2015.
- [7] M. Y. Tsalamalal, J. C. Martin, M. Ammi, A. Tapus, and M. A. Amorim. Affective handshake with a humanoid robot: How do participants perceive and combine its facial and haptic expressions? In *2015 International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 334–340, Sept 2015.
- [8] Chao Zhai, Francesco Alderisio, Piotr Słowiński, Krasimira Tsaneva-Atanasova, and Mario Di Bernardo. Design of a virtual player for joint improvisation with humans in the mirror game. 11:e0154361, 04 2016.