Analysis of Tactile Instructions Used in the Interaction with a Humanoid Robot

Fransiska Basoeki* Fabio DallaLibera† Takashi Minato‡ Hiroshi Ishiguro* ‡

Abstract

Touch can be an intuitive mean of communication to teach motions to humanoid robots. However, the way inexperienced users use touch to communicate was not sufficiently investigated. In this work we analyze how inexperienced users employ tactile instructions for teaching.

1 Introduction

Humanoid robots are increasingly becoming popular and more people are using it either for hobby or research or even for advertisement. In order to develop motions for these robots, users usually need to specify the positions the robot should assume during the movement. This is mostly done by setting each motor's target position using sliders in a control interface [13]. A typical interface for this is shown in figure 1. Motions are developed by utilizing *keyframes*, ie. important points in between smooth transitions of a movement. In other terms, the user needs to specify the complete robot posture for a set of keyframes, the sequence of which is then interpolated. It becomes necessary for the user to decide the right joints, the right directions and the right angles to move in order to obtain the desired posture. Other, more advanced methods such as motion capture and retargeting were presented [8], but they are often laborious and require special devices which demand high costs.

Among many ways of communication between humans, touch is an intuitive way of human-human interaction. Just by touching, various information, such as perceptions, thoughts, or feelings, are able to be conveyed to others [5, 2]. Touch is also often used by instructors in sport or dance classes to adjust the student's posture or motion [12].

As how touch provides an intuitive way of communication between humans, it can also be an attractive way to interact with robots. There have been some researches on implementing tactile sensing on robots as a way to interact with humans [10], particularly on robot manipulators [3]. Kinesthetic demonstration was largely employed with humanoid robots as well, for instance, teaching a robot some arm movements of a manipulation task [4].

Users have their own image in their mind of how they want the robot to move, and they can try to communicate it through touch. In order for the communication to be successful the robot must be able to correlate these touches to motor rotations. The interpretation is generally a complex problem. For instance, the meaning of the same touches could differ depending on the context they were given at. As an example, a touch on the upper part of the leg when the robot is standing could mean that the robot must bend the leg backwards, while when the robot is squatting, the same touch on the leg could mean that the robot should bend its knee further. This is illustrated in figure 2.

Apart from the robot's context, the users' individual ways of touching could result in ambiguities in the interpretation of the meaning of tactile instructions. To develop a better system for the interpre-

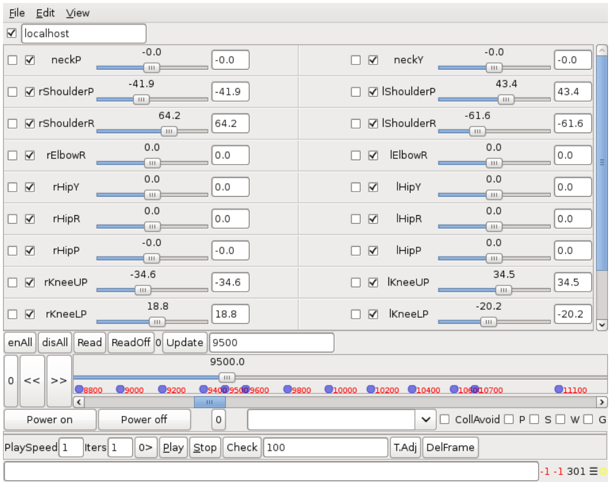


Fig.1 A typical slider interface. The angle of each joint is directly controlled by operating a slider.

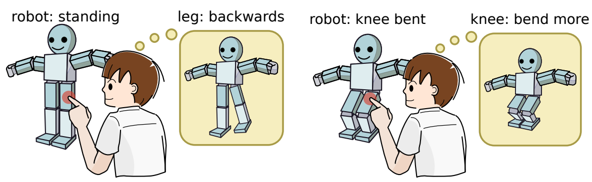


Fig.2 Different touch meaning in different contexts. The same touch corresponds to two different intentions depending on the context.

* Osaka University
† Padova University
‡ Asada Project, ERATO, JST

tation of the meaning of touches we need to investigate how inexperienced users use touch to provide teaching instructions. Once the basic teaching policies employed by humans is understood, it will be possible to provide the robot with intuitive touch interpretation algorithms.

We can expect the way of teaching to be dependent on the task, ie. the motion being taught. We can also imagine different users to use different level of abstraction in providing their will. For instance users familiar with simple devices could find natural and effective to adhere to a strict association between touch sensors and changes in the joint angles of the robot. Conversely other users could expect the robot to execute a high number of related motor changes given a single tactile instruction. This user dependency in the touch instruction interpretation does not anyway mean that no recurring feature in the mapping between touch instructions and their interpretation is to be expected. In fact, it would be unlikely that the users find natural to use completely arbitrary mappings, as it can be understood observing that people are able to use touch to communicate with other people.

This paper presents a preliminary study on a single subject that shows how analysis of tactile instructions can be used to provide insights on the features of the mapping between touches and their meaning. In the experiment conducted, the user interacts with a humanoid robot and develops a motion using exclusively touch instructions. The robot responds to the user's touches by using a database of examples of the mapping between touch instructions and joint movements. This database is initially empty and the robot has no knowledge on the meaning of tactile instructions. However, during motion development, the user can teach the robot the meaning of touch instructions by direct manipulation. Once a new meaning is taught, this is stored in the database and used for touch interpretation.

The data collected in our experiment support the hypothesis that the mapping between tactile sensors and the movement that should be executed is not straightforward. Interestingly, however, the data provided by the subject appear to suggest that the way the robot should respond to touch instructions can be described using a low-dimensional subspace of the joint space. As widely known [6, 9] motions can often be described in a low-dimensional subspace of the joint space as well. Interestingly, in our case study, the low-dimensional subspace used for describing the motions can be used to describe the movements that should be performed in response to tactile instructions as well. This indicates the possibility of employing knowledge on the motion being developed (for instance the keyframes provided) to improve the interpretation of tactile instructions.

The details of the interface and the algorithms used for exploiting the database of examples will be outlined in section 2 and section 3 respectively. The experiment and the results obtained will be discussed in section 4. Finally, section 5 will conclude by proposing future works.

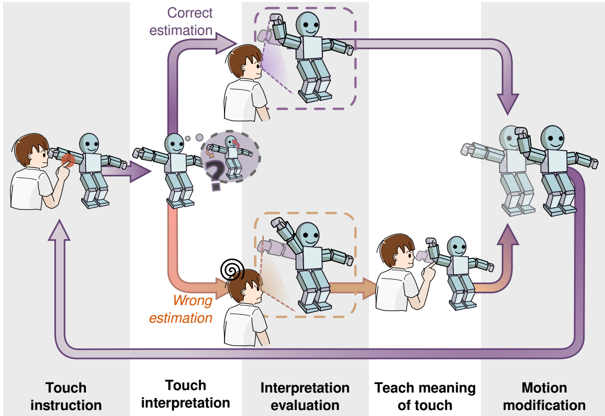
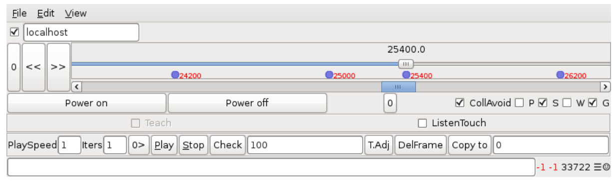


Fig.3 Schema of the motion development. Users touch the robot to modify the posture of a keyframe. The robot interprets the meaning of the tactile instruction and modifies its posture accordingly. Users then evaluate the motor position change. If the movement corresponds to their intention then they will continue to develop the motion, otherwise they will teach the robot the meaning of the touch instruction.



(a) Touch interface

Fig.4 The interface for motion development, consisting essentially of a single timeline and buttons for playing and stopping the motion execution.

2 Interface

Our proposed teaching by touching interface is able to seamlessly switch between two modes, schematized in figure 3.

In general, the robot will always be standby in *motion development mode*, ie. it will wait for the users' touch patterns and move according to its knowledge of the touch protocol. By using this mode users are able to create keyframes through the interface shown in figure 4.

In particular, users select a certain instant of the motion using a timeline and then, using tactile instructions, they are able to set the posture that the robot should assume at that time using tactile instructions.

When the robot does not understand the meaning of the touch pattern, or when the robot does not move according to what the user intended, the operator is able to switch to the second mode, the *teaching the meaning of touch mode*.

During this teaching mode, users are able to show the robot the intended movement by kinesthetic

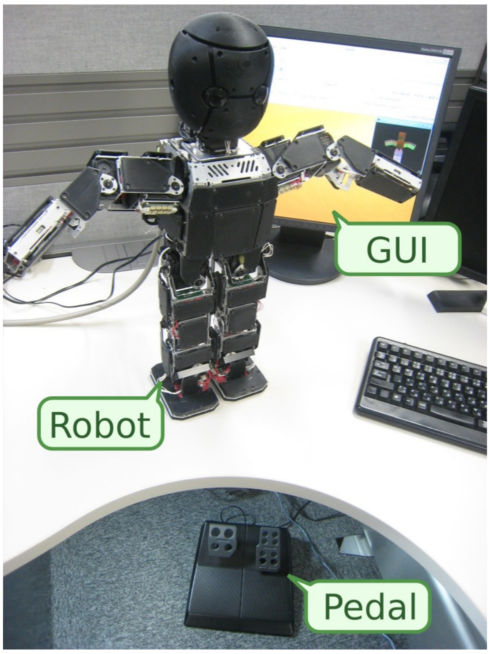


Fig.5 Experimental setup. A pedal device, commonly used for racing game, allows users to easily switch between the "motion development mode" and the "teaching the meaning of touch" mode without having the need to move his hand away from the robot to click a button on the interface.

demonstration, ie. they can move the robot to the posture the robot should have gone by the provided touch instruction. Expressly, during this mode when the sensor on both sides (palm and top) of one hand are pressed the motors of that arm are switched off so that the arm can be moved freely. When the sensors are released, the power will be turned back on while maintaining the position where the limb was moved to. Similarly, the motors of each leg can be turned off by pressing the top and bottom side of the corresponding foot and the motors of the head can be powered off by pushing the front and back sensors of the head.

When the user is satisfied with the change of pose, she or he switches back to the "motion development mode" and the robot stores the association between the touch pattern given at the beginning of the "teaching the meaning of touch mode" and the movement shown during the teaching mode.

Figure 5 depicts the experimental setup. The system includes a pedal that allows the user to change between the two modes, the "motion development mode" and the "teaching the meaning of touch" while keeping touching the robot. Precisely, the "teaching the meaning of touch" is activated when the pedal is pressed and terminated when the pedal is released.

3 Algorithm

As introduced before, touch is considered to be context dependent, therefore a mapping between a touch pattern and a context to a joint modification is essential. The context elements currently considered are:

- the current robot posture, ie. the position of the motors, as a different posture can create a different context (Fig. 2);
- the robot's orientation, estimated from the accelerometer sensors reading, since for instance standing and lying down can correspond to different contexts;

To produce the mapping, the k-Nearest Neighbor algorithm with $k = \infty$ is used. During the motion development, the user provides examples of input (touch pattern and context) I_i and output (joint modification vector) M_i. By using the distance in the high dimensional space between the system input I_* and each of the example input I_i, a weighting to get the output vector M_* can be obtained. Precisely, denoting by E the number of collected examples, the system output is calculated as

$$M_* = \sum_{i=1}^{E} \omega_i M_i \tag{1}$$

It was observed that using a decreasing function of the Euclidean distance for the weights gives bad performance in estimating the desired joint modification. Therefore, a different weighting function was defined. To begin with, touch information should have more importance than the context. For instance, suppose the user taught solely arm motions and there is no knowledge on the leg. Pushing the sensors on the leg should usually not cause any arm movements even if the context is very similar to one of some examples that include arm motions. To avoid such situations, the weightings for examples which include pressure of sensors not pressed in the current input I_* is set to 0, i.e. $\omega_i = 0$.

Next, when the pressure of a sensor with force f corresponds to a single motor joint change, the user would assume that pushing with less strength would cause less change in the joint, while, on the opposite, pushing with greater force would correspond to a larger change. However, any system, where the weighting is based on a decreasing function of the distance between the examples and the current input would behave differently. Expressly, when the force applied to that sensor is different from the force f, either greater or smaller, the calculated output would always result in a smaller angle change.

To solve this problem, the weight ω_i can be obtained by defining it as the product of two terms, one that increases linearly with pressure and one that decreases as the current input I_* and the i-th example input I_i differ. More formally, let us define

- $F_i\,[s]$, the force of the s-th sensors in the i-th example, where $1 \le s \le n$
- P_i, the joint angles of the robot in i-th example,

- O_i, the value of the accelerometers along the three axis, which provide indication on the orientation of the robot in the i-th example,

where $1 \leq i \leq E$. We then assume analogous definitions for the current input's I_* . The weight ω_i is then calculated as:

$$\omega_i = \begin{cases} 0 & if\ \bigvee_{s=1}^{n} (F_i[s] > 0) \wedge (F_*[s] = 0) \\ \alpha_i \cdot \beta_i & otherwise \end{cases} \tag{2}$$

where

$$\alpha_i = \prod_{s:F_i[s]>0} F_*[s] / F_i[s] \tag{3}$$

$$\beta_i = \frac{1}{1 + \sqrt{\gamma_i}} \tag{4}$$

$$\gamma_i = \sum_{s:F_i[s]=0} F^*[s]^2 + \|P_* - P_i\|^2 + \|O_* - O_i\|^2 \tag{5}$$

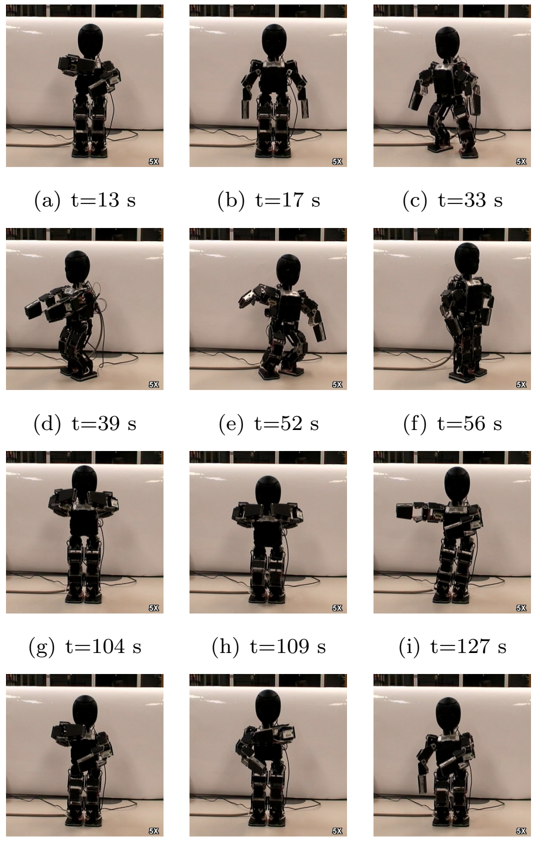
Equation 4 was derived based on practical experiments where this was observed to give the most intuitive behavior among several decreasing functions that were tested. Further details on the algorithm can be found in [1].

4 Experiment and Results

In the experiment, the user was asked to teach a humanoid robot a motion using the proposed touch interface. The motion is based on the first half of *Algorithm Exercise* (アルゴリズム体操). *Algorithm Exercise* is a body exercise aimed at small children[*]. The motion involves changing facing direction and simple hand movements. The main keyframes of the motion are shown in figure 6. This motion was chosen because on the one hand it is complex enough to require the user to teach a high number of different postures and on the other hand it is simple from the view point of robot balancing.

The robot employed for the experiment is *M3-Neony*, a 22 degrees of freedom humanoid robot (see figure 7) equipped with 92 tactile sensors (see figure 8), accelerometers along three axis and two gyroscopes[†]. The experiment setup is as shown in figure 5.

The user is this case study is a 23 year old Japanese right handed male with no experience on the teaching by touching interfaces. During the motion development the user provided the meaning of 156 tactile instructions. In total the user touched the robot 672 times. Once taught, multiple instructions can be reused simultaneously, formally speaking, several weights ω_i can be non-zero. Data show that the user exploited this fact, in fact the total number of touches instructions provided to the robot, counting the superposition of n tactile instructions as n, is 1322.



(a) t=13 s (b) t=17 s (c) t=33 s
(d) t=39 s (e) t=52 s (f) t=56 s
(g) t=104 s (h) t=109 s (i) t=127 s
(j) t=173 s (k) t=175 s (l) t=183 s

Fig.6 Robot's Algorithm Exercise movement. Shown time is the robot's actual movement timing in seconds. Robot's motion time is 5 times slower than the actual Algorithm Exercise timing.

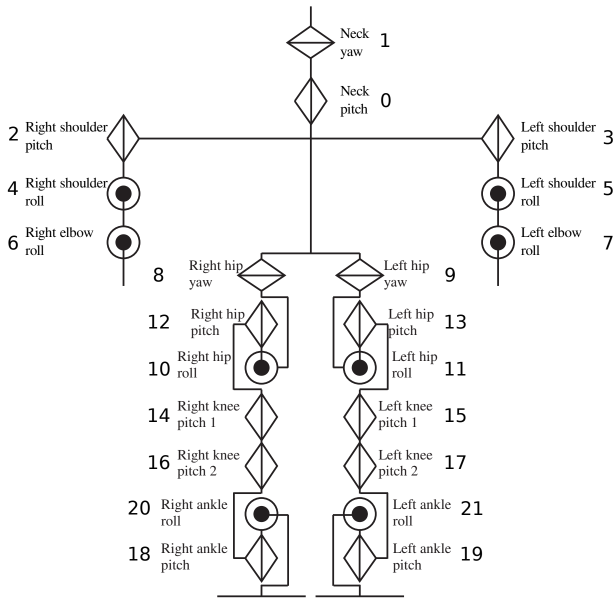


Fig.7 Diagram of the degrees of freedom of the robot.

[*] Video is available at `http://robotics.dei.unipd.it/~fabiodl/video.php?algo` (*Motion developed by touch*)
[†] Further details on this robotic platform can be found in [7].

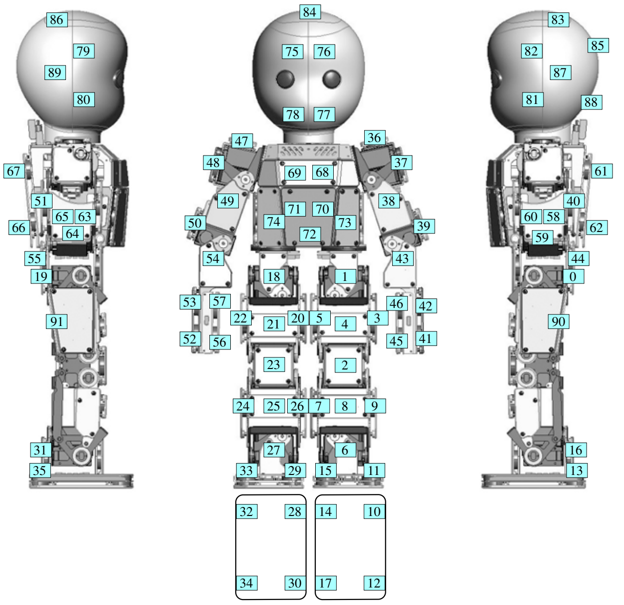


Fig.8 Diagram of the tactile sensors placement

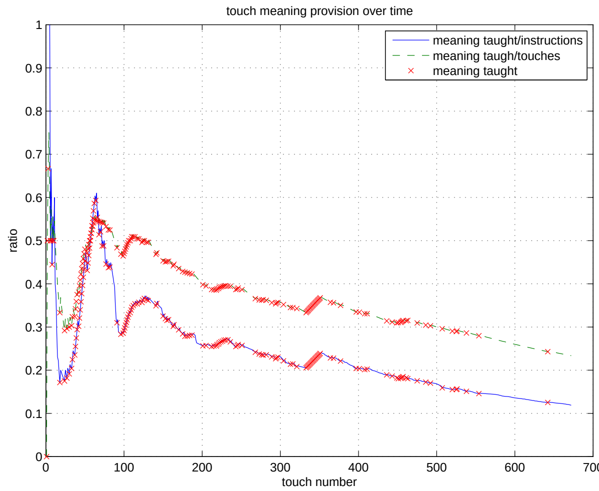


Fig.9 Touch meaning provision over time. Red crosses indicate when the meaning of the touch was taught to the robot. The continuous blue line indicates the ratio between the touch meanings taught and the number of touches applied to the robot. The dashed green line indicates the ratio between the number of touch meanings taught and the number of touch instructions given. The initial strong decrease and subsequent increase of the ratios is due to the fact that the user firstly taught the meaning of a touch instruction, reused it several times and then provided the meaning of a new set of tactile instructions.

As explained before, when a tactile instruction is not understood the user can teach the robot the meaning of such instruction for further use. In this way the robot's mapping is enriched and refined continuously. Figure 9 shows that the user needs to teach less and less instructions over time because he can effectively reuse the ones already taught. In particular, the ratio between the touch meanings taught and the number of touches provided is displayed. As previously stated, since a single touch can combine several taught instructions, the ratio between the number of touch meanings taught and the number of instructions provided decreases even faster.

In order to study the mapping between touch sensors and motor joint changes, we analyzed the examples of mapping between touch instructions and motor posture changes taught by the user. In particular we calculated the mutual information between each of the sensors and the rotation given for each of the motors. To compute the mutual information, we initially discretized the data. Each sensor information was set to 0 if its value was less than 20% of the maximum force measurable by the sensor and 1 otherwise. Each motor change information was set to 0 if the user moved the motor less than 6 degrees, to -1 if the user moved the motor more than 6 degrees in clockwise direction and +1 if the user moved the motor in counter clockwise direction more than 6 degrees.

Let us denote the probability that the s-th sensor value to be σ as $p_s(\sigma)$. The probability $p_s(\sigma)$ was estimated from the collected as $p_s(\sigma) = |F_i\,[s] = \sigma|\,/E$, where E is the number of provided examples and $|F_i\,[s] = \sigma|$ denotes the cardinality of the set of collected examples where the sensor s assumed value σ. Due to our discretization σ can assume only the values 0 and 1. Similarly let us denote by $p_m(\mu)$ the probability that a touch instruction corresponds to a change in the position of the m-th motor of μ. Formally we estimated $p_m(\mu)$ by setting $p_m(\mu) = |M_i\,[m] = \sigma|\,/E$, where $M_i[m]$ denotes the position change for the m-th motor provided in the i-th example. Finally let us denote by $p_{s,m}(\sigma,\mu)$ the joint probability of the s-th sensor value being σ and the change of the m-th motor being μ for the same example. The mutual information between a sensor s, $1 \leq s \leq 92$, and the change of the position of a motor m, $1 \leq m \leq 22$, was computed as

$$\hat{I}_{s,m} = \sum_{\sigma \in \{0,1\}} \sum_{\mu \in \{-1,0,1\}} p_{s,m}(\sigma,\mu) \cdot \log_2 \left(\frac{p_{s,m}(\sigma,\mu)}{p_s(\sigma) \cdot p_m(\mu)} \right)$$

The normalized mutual information $I_{s,m}$ was then computed considering the entropies [11], expressly

$$H_s = - \sum_{\sigma \in \{0,1\}} p_s(\sigma) \cdot \log_2 p_s(\sigma)$$

$$H_m = - \sum_{\mu \in \{-1,0,1\}} p_m(\mu) \cdot \log_2 p_m(\mu)$$

$$I_{s,m} = \frac{\hat{I}_{s,m}}{\sqrt{H_s \cdot H_m}}$$

Figure 10 illustrates the results. We notice that mainly the user touched the sensors on a limb to move motors on the same limb. However, there is no one to one correspondence between joints and sensors. Several sensors are used to actuate the same joint and conversely the same sensor actuates several joints. We can observe also correlation between the sensors on the top of the head and the leg motors. Direct inspection of the touch data show that this was done to teach the robot to squat when it is touched on the head. Even more interestingly, we can notice a correlation between the sensors places on the side of the robot's body and the corresponding leg. Direct inspection shows that in these cases the user employed the sensors on the side to tell the robot to rotate the corresponding leg and bring the knee outwards on that side (see figures 6(c) and 6(e)). Correlation between the sensors on the upper part of the left leg ($s00.lHipB$ and $s01.lHipF$) and motors of the right leg ($m10.rHipP$) also emerges, since often the two legs are moved together in order to maintain both feet parallel to the ground.

Multiple motors are usually moved with a single touch instructions. It is hence interesting to observe if there are consistencies in the relationship between the motor changes of different motors in the set of changes provided as meaning of touch instructions. Computing the mutual information between couples of motors we obtained the data reported in figure 11. We notice a very strong correlation between motors that belong to the same limb. Correlation between the two legs is also observable, in particular between the pitch joints of the hips ($m10.rHipP$ and $m17.lHipP$) and of the ankles ($m13.rAnkeP$ and $m20.lAnkleP$).

This high correlations suggest that the motor changes given by the user could actually be located in a low-dimensional manifold of the 22-dimensional motor space. This fact could be exploited in the estimation of meaning of touch instructions. For simplicity we focused on linear subspaces as possible manifolds and we analyzed how much information is lost when projecting the motor change given for the e-th example on a low-dimensional subspace constructed using the first $e-1$ touch examples. More precisely we took the motor changes specified in the first $e-1$ examples $M_1 \dots M_{e-1}$, subtracted the mean and applied Principal Component Analysis (PCA). We then projected the e-th example motor change M_e on the subspace defined by the first q principal components $v_1 \dots v_q$, and observed the infinity norm of the reconstruction error:

$$\epsilon_q(e) = \left\| M_e - \sum_{i=1}^{q} M_e^T v_q v_q \right\|_\infty$$

Figure 12 reports the reconstruction error for different settings of q, $1 \le q \le 22$, averaged over all the examples $e = 1 \dots E$. For comparison, the error obtained by applying PCA on the whole set of examples, ie. $M_1 \dots M_E$ is also reported. We notice that the difference between the reconstruction error obtained using just the first $e-1$ examples is not much higher than the one obtained using the whole data set $M_1 \dots M_E$.

As stated in the introduction, it is well known that for many tasks the complete movement of the robot lies on a small subspace as well. It is therefore interesting to analyze whether the subspace where motions can be projected with little errors is related to the motor changes desired as response of touch instructions. For each of the E examples provided by the user we collected all the postures that the user brought the robot to before teaching the M_e. For each of these sets of postures we then subtracted the mean and applied PCA to determine the principal components $\bar{v}_1 \dots \bar{v}_q$. The average reconstruction error norm obtained projecting the touch examples M_e

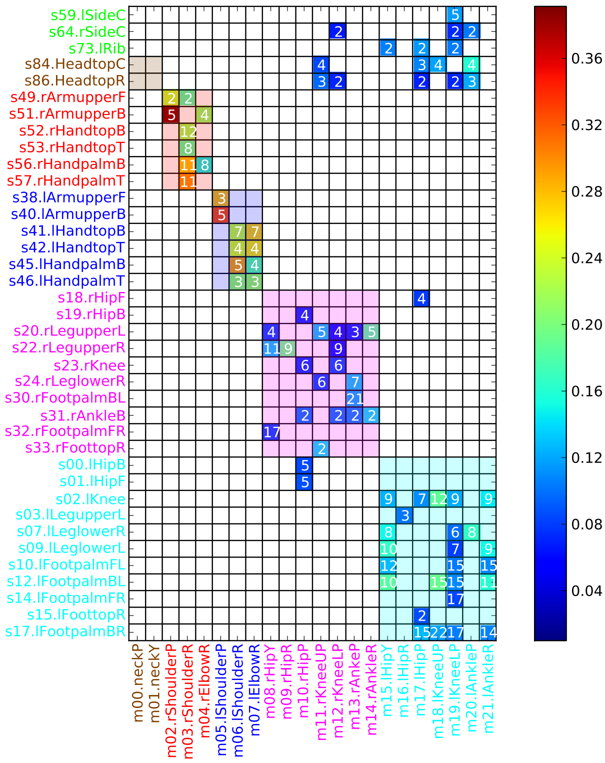


Fig.10 Mutual information between sensors and motor angle change. Each row represents a sensor and each column a motor. The color of the intersection indicates the normalized mutual information value. For clarity only mutual information values higher than 0.01 are indicated, and only sensors that have mutual information value higher than 0.01 with at least one motor are reported. The number inside each cell indicates the number of touches for which the corresponding sensor was pressed and the corresponding motor was moved. Different colors of the labels are used to indicate different robot parts. The areas of the map that correspond to sensor and motors of the same robot part are highlighted by a rectangle of the corresponding color.

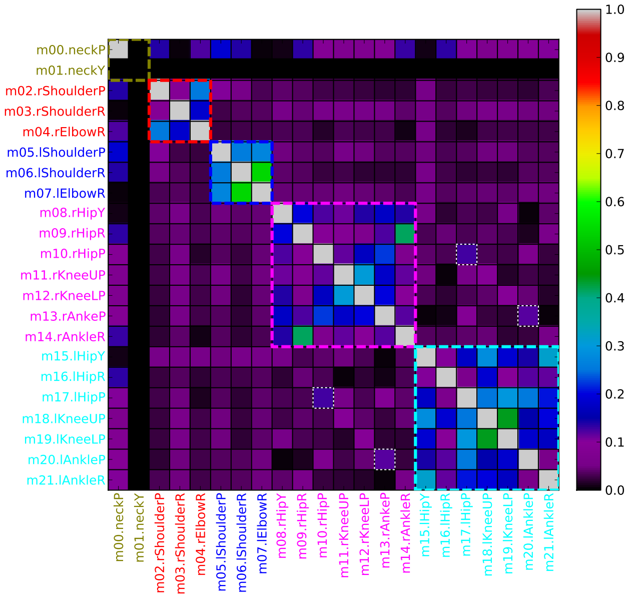


Fig.11 Mutual information between couples of motors. The color of the intersection indicates the normalized mutual information value. Different colors of the labels are used to indicate different robot parts. The areas of the map that correspond to motors of the same robot part are highlighted by a dashed square of the corresponding color. High correlations between corresponding joints of the two legs are highlighted by a white dotted line.

on the subspace of dimension q defined by $\bar{v}_1 \ldots \bar{v}_q$ is also reported in Figure 12. We notice that except for low values of q ($q < 5$) the reconstruction error is comparable to the one of the projection on the subspaces constructed using the motor change information. These preliminary results, that need more intensive verification, seem to suggest that users tend to provide desired changes consisting in movements that lie in the subspace defined by the motion they want to develop. This fact could be exploited to improve the touch interpretation given the knowledge of the frames set by the users during the motion development.

At a first glance it might appear very strange that the motor angle changes can be projected on a subspace extracted using posture angles. However, it must be considered that actually the postures can be thought as a linear combination of motor position changes from the initial, zero position.

The interesting fact is that the user in our case study kept the meaning of touch instructions inside the subspace defined by the motion, and did not essentially provide motor changes in the subspace orthogonal to the space in which the motion lies. In

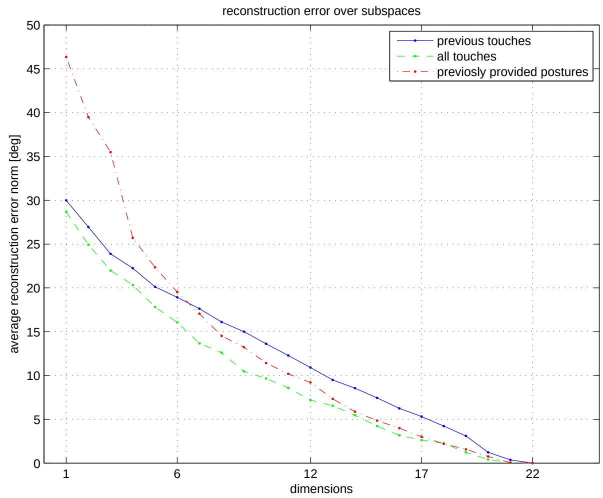


Fig.12 Reconstruction error for different number of dimensions of the subspace.

other terms, the user was requested to develop a motion, "Algorithm exercise", and developed it by defining a set of postures that lie in a certain subspace of the motor space. When setting these postures that realize the task, instead of setting them in a completely free manner using motor changes in the whole motor space, he restricted his instructions to movements similar to the ones that compose the target motion.

This peculiarity could derive from the fact that for the task chosen gestures are more important than the actual postures taken by the robot. For instance, imagine the robot task to consist of grasping an object and rising it as high as possible. We can imagine the robot motion to lie on a subspace that makes the robot arms move vertically. However users would probably concentrate their instructions in adjusting the hands distance in order to achieve an adequate grasp, and would therefore probably provide a high number of instructions in a subspace orthogonal to the one of the motion. Future works will need to include the analysis of data from different types of tasks in order to verify this hypothesis.

5 Conclusions and Future Work

Touch instructions can provide an intuitive way of interacting with humanoid robots, at least in the field of motion developments. In human-human interaction, in fact, it is natural for a coach to use touch to show trainees how to modify their posture. However, interpretation of unconstrained touch instructions reveals to be complicated.

A possible approach consists in analyzing the strategies utilized by humans to naturally express how they intend to modify a humanoid robot motions, derive general rules and use them to make the robot perform a suitable interpretation of touch instructions. While we can expect user dependencies in the ways of teaching, it appears likely that users share some basic features of the mapping between touch instruction and motion modification.

This paper reports a preliminary analysis of the data provided by a single user. The results show that, as expected, there is no simple correspondence between tactile instructions and desired motor changes, and therefore specific algorithms for interpretation of touch instructions are required. Some general tendencies appear however to emerge even from the analysis of this single case, encouraging further studies on the topic.

Data collected suggest that usually a limb is moved by touching touch sensors on the same limb, as could be expected. Interestingly this appears however not to be true when the users want to convey higher level behaviors to the robot. In the case considered in this paper, for instance, the subject touched the head of the robot to make it squat or touched its side to express the desire to turn the leg and bring the knee outwards.

Further analysis of the collected data show that usually the posture modifications desired by the users lie in a (linear) subspace of the motor space. Interestingly this subspace seems to be highly correlated to the subspace where the motion itself lies. This fact suggests us that the keyframes of the motion that the user is developing could be used to improve the estimation of the meaning of touch instructions.

Future works will need to consider a higher number of subjects, to observe whether the findings of this paper can be generalized or are constrained to the task and the subject of this experiments. Furthermore, other ways of teaching can be expected for different users. For instance we could imagine some users to provide higher level instructions that map single tactile patterns to complete whole body "actions". Conversely we could imagine other users to assume low capabilities for the robot and adopt a very strict mappings between sensors and motors.

References

1) Fabio DallaLibera, Takashi Minato, Ian Fasel, Hiroshi Ishiguro, Enrico Pagello, and Emanuele Menegatti. A new paradigm of humanoid robot motion programming based on touch interpretation. *Robotics and Autonomous Systems*, 57(8):846–859, 2008.
2) Tiffany Field. *Touch.* Bradford Books, 2003.
3) G. Grunwald, G. Schreiber, A. Albu-Schffer, and G. Hirzinger. Touch: The direct type of human interaction with a redundand service robot. In *Proceedings of the 10th IEEE Int. Workshop on Robot and Human Interactive Communication (ROMAN 2001)*, Bordeaux/Paris, France, 2001.
4) M. Hersch, F. Guenter, S. Calinon, and A. Billard. Dynamical system modulation for robot learning via kinesthetic demonstrations. *IEEE Trans. on Robotics*, 24(6):1463–1467, 2008.
5) Matthew J. Hertenstein. Touch: Its communicative functions in infancy. *Human Development*, 45:70–94, 2002.
6) Auke Jan Ijspeert, Jun Nakanishi, and Stefan Schaal. Learning attractor landscapes for learning motor primitives. In Suzanna Becker, Sebastian Thrun, and Klaus Obermayer, editors, *NIPS*, pages 1523–1530. MIT Press, 2002.
7) Takashi Minato, Fabio DallaLibera, Satoshi Yokokawa, Yutaka Nakamura, Hiroshi Ishiguro, and Emanuele Menegatti. A baby robot platform for cognitive developmental robotics. In *Workshop on "Synergistic Intelligence" at the 2009 IEEE/RSJ International Conference on Intelligent RObots and Systems (IROS 2009)*, St. Louis, USA, 2009.
8) Shinichiro Nakaoka, Atsushi Nakazawa, Kazuhito Yokoi, Hirohisa Hirukawa, and Katsushi Ikeuchi. Generating whole body motions for a biped humanoid robot from captured human dances. In *2003 IEEE International Conference on Robotics and Automation (ICRA 2003)*, pages 3905–3910, Taipei, Taiwan, 2003.
9) Atsushi Nakazawa, Shinichiro Nakaoka, and Katsushi Ikeuchi. Synthesize stylistic human motion from examples. In *2003 IEEE International Conference on Robotics and Automation (ICRA 2003)*, pages 3899–3904, Taipei, Taiwan, 2003.
10) W. D. Stiehl, L. Lalla, and C. Breazeal. A somatic alphabet approach to sensitive skin for robots. In *2004 IEEE International Conference on Robotics and Automation (ICRA 2004)*, pages 2865–2870, New Orleans, USA, 2004.
11) Alexander Strehl and Joydeep Ghosh. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.*, 3:583–617, 2003.
12) Takahiro Takeda, Yasuhisa Hirata, and Kazuhiro Kosuge. Hmm-based error recovery of dance step selection for dance partner robot. In *2007 IEEE International Conference on Robotics and Automation (ICRA 2007)*, pages 1768–1773, Roma, Italy, 2007.
13) Takenori Wama, Masayuki Higuchi, Hajime Sakamoto, and Ryohei Nakatsu. Realization of tai-chi motion using a humanoid robot. In René Jacquart, editor, *IFIP Congress Topical Sessions*, pages 59–64. Kluwer, 2004.