Teaching Motions by Touching

Fabio DallaLibera*, Takashi Minato[*], Hiroshi Ishiguro[*+], Enrico Pagello*, Emanuele Menegatti*
*Department of Information Engineering, University of Padova, Padova, Italy, dallalib@dei.unipd.it
[*]ERATO, Japan Science and Technology Agency, Osaka, Japan
[+]Department of System Innovation, Osaka University, Osaka, Japan

Summary

Literature presents several examples of employment of touch in human-robot interaction, dating back to teaching by playback of robotic arms or interpretation of tactile gestures[1]. Recently kinesthetic demonstration has drawn particular attention in the field of humanoid robots[2]. In these works the robot is just a passive entity. Conversely, in our works[3,4] the robot responds actively to touch instructions by interpreting the touch meaning and moving its motors accordingly, more similarly to what happens in human-human communication.
We showed that the intuitiveness of touch can be exploited to allow inexperienced users to teach motions to humanoid robots. Precisely in our first works[3] we used a simple key-frame based representation for the motion description and focused our attention on the meaning of touch. Practically, the user watches the robot executing the motion, chooses an instant in time when the motion should be modified and touches the robot to adjust the robot posture at that time. The robot responds to the pressure on its sensors by changing its joint angles in accordance to its interpretation of the touch meaning. When the robot fails to interpret the meaning of the touch the user teaches by another way of communication how she or he wanted the robot to move. Expressly when the robot fails to understand the meaning of a touch instruction in our implementation the user can teach the robot the desired movement associated to the touch either by direct manipulation or by a classical slider based interface. This can be restated in machine learning terms: examples of touches and corresponding joint angle changes given by the user are used by a supervised learning algorithm to interpret the meaning of touches and online provision of new examples allows refining the mapping where the robot fails to interpret the user intention.
The system shown good performances in terms of reduction of motion development time with respect to classical, slider based editors. Analyzing the data acquired during the touch interaction also allowed us to get insights on the way humans use touch to convey information. For instance, we identified context elements that influence the meaning of touch and highlight strong user dependence in the way of teaching. In particular preliminary results seem to suggest that different users employ different levels of abstraction when using touch to communicate their intended posture modification.
Our second series of works[4] switched the focus from touch interpretation to more advanced forms of representation of the motion, and expressly we used a specifically designed Central Pattern Generator (CPG) to represent the motion and used touch to set its parameters. The developed system allows to develop periodic motions simply by touching the robot, without the support of any other GUI or any alternative protocol. We showed the feasibility of the approach by developing from scratch in a short time three periodic motions, namely walking, side stepping and crawling.
Videos are available at *http://robotics.dei.unipd.it/~fabiodl/video.php?videoGroup=humanoids09tactile*

---

[1] R. Voyles and P. Khosla, Tactile Gestures for Human/Robot Interaction, 1995 IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems(IROS 1995), Vol. 3, pp. 7-13, Pittsburg, USA, 1995

[2] M. Hersch, F. Guenter, S. Calinon and A. Billard, Dynamical System Modulation for Robot Learning via Kinesthetic Demonstrations, IEEE Trans. on Robotics, Vol. 24, No. 6, pp. 1463-1467, 2008

[3] Fabio Dalla Libera, 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, Vol. 57, No 8, pp.846-859, Jul, 2009

[4] Fabio Dalla Libera, Takashi Minato, Hiroshi Ishiguro and Emanuele Menegatti, Direct Programming of a Central Pattern Generator for Periodic Motions by Touching, Robotics and Autonomous Systems, Special Issue on Advances in Autonomous Robots for Service and Entertainment, 2009 (to Appear)

Motivation

Humanoid robots are becoming a more and more easily available entertainment device on the market, at a lower and lower cost. In most of the interfaces for their programming the target position of each of the servomotors is specified by the user using a slider[5] for some time instants, normally called keyframes. The robot control board then interpolates the keyframes to generate the motion. The motion development process therefore requires the user to identify, for each posture, which are the parts of the robot that should be moved, realize which are the joints, along the kinematic chain, that cause the desired movements, and determine, for each of the joints, the right rotation direction and the appropriate rotation amount. Although other techniques, such as motion capture and retargeting[6], can be employed for motion development, these methods are still cumbersome, and require the use of expensive devices.

Touch is an intuitive method of communication. It is employed in human-human interaction, for instance by sports coaches or dance instructors[7] to correct a learner's posture or motion. Tactile interaction therefore appears particularly appealing as an intuitive method to teach to humanoid robots as well.

Although very intuitive for humans, touch interpretation is not straightforward since similar touches could have different meanings depending on the context. For instance, if the robot is standing, touching the upper part of one leg could mean that the leg should bend further backwards. However, if the robot is squatting, the same touch could mean that the robot should move lower to the ground by bending its knees (see Fig. 1).

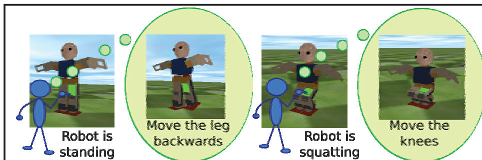

Fig. 1 Example of the context dependence of the touch instruction meaning

Results

We devised an algorithm for the interpretation of the touch meaning based on k-Nearest Neighbor with a specifically devised weighting schema.

This algorithm takes as input a touch pattern (and its context, consisting in the robot's posture, its orientation in the space and the velocity of its center of mass) and gives as output the expected desired joint angle change. As briefly reported in the summary, the algorithm is trained using examples of the mapping collected during the motion development.

Analyzing data collected during the experiments, we showed that the touch interpretation cannot be explained by linear models. In particular employing two datasets, collected during the development of two different motions (walking and jumping) we showed that linear models tend to overfit the data. Table 1 provides the comparison of the errors in the prediction of the desired angle change for the two algorithms under different settings of the training and test datasets.

Training dataset	Test dataset	Average relative error	
		linear regression	K-NN
JUMP	JUMP	0.1872	0.1863
JUMP	WALK	715.0779	1.0325
WALK	JUMP	2.6658	0.9569
WALK	WALK	3.98E-06	0.1022

Table 1 Errors of the k-NN algorithm and linear regression.

[5] T. Wama, M. Higuchi, H. Sakamoto, and R. Nakatsu, Realization of tai-chi motion using a humanoid robot, IFIP Congress Topical Sessions, pp. 59-64, 2004.

[6] S. Nakaoka, A. Nakazawa, K. Yokoi, H. Hirukawa, and K. Ikeuchi, Generating whole body motions for a biped humanoid robot from captured human dances, 2003 IEEE Intl. Conf. on Robotics and Automation (ICRA2003), pages 3905-3910, Taipei, Taiwan, 2003.

[7] T. Takeda, Y. Hirata, and K. Kosuge, Hmm-based error recovery of dance step selection for dance partner robot., 2007 IEEE Intl. Conf. on Robotics and Automation (ICRA 2007), pp. 1768-1773, Roma, Italy, 2007.

We then divided the collected examples in classes, where each class indicates whether in the example output each motor was rotated in clockwise direction, counterclockwise direction or was not moved. We ran Quinlan's C4.5 algorithm and discovered that the features of the context that appear in the highest level of the decision tree, i.e. the features that more strongly influence the meaning of touch are given by the rotation of the joints near the torso. This makes sense, since those joints determine the overall positions of the limbs. Table 2 reports for each joint the levels at which its angle appears in the conditions of the decision tree generated by the C4.5 algorithm (the joint IDs are visible in Fig.2).

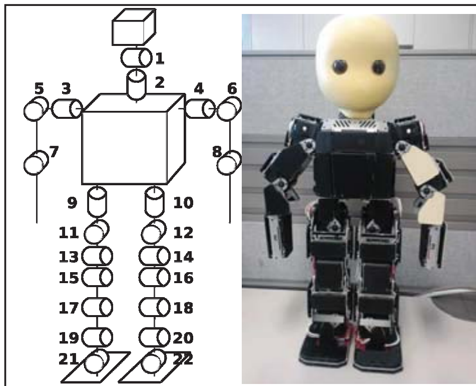

Fig. 2 Employed robot and its degrees of freedom.

Joint	Dataset	
	JUMP	WALK
1	2	
2	17,21,22	9
3	3	3
4		5,11
6		2
7	20	
8	5,8,19	
9	1	3
11		10
12	0	1
14		3
15		2,8,10
16	5	3,4

Table 2 Levels at which each joint appear in the branchings of the decision trees constructed for the two datasets.

The jumping motion was developed both with the proposed, touch based system and with a classical slider based editor. As a result, the same user realized the motion in 17 minutes with our system while using a classical, slider based interface he spent over 40 minutes.

We then observed that the meaning associated to touch is very user dependant, and in detail that different human operators use a different level of abstraction when providing their touch instructions. In an experiment with six subjects we noticed four types of teaching:

1. a nearly fixed mapping from a small set of sensors to the joints; the context has little or no influence (users B and D);
2. a more free mapping, that uses several sensors to move the same joint (users C and F)
3. a mapping based on physical considerations, i.e. the joints are imagined to be elastic; in this case, the context, for instance the position of the ground, becomes crucial (user A)
4. a very high level representation of the motion, where for instance just the limb that should be moved is indicated by touching (user E)

These qualitative observations, obtained by direct inspection of the mapping between pressed sensors and moved joints, seem to be supported by quantitative results.

More precisely, we trained the K-NN algorithm with the data of a single user, and calculated the expected outputs (joint angle changes) on the touch patterns provided by all the other users. We then calculated the average correlation between the outputs given the same input for each couple of users, and converted this correlation to a distance (see [3] for the details). Finally we applied the multidimensional scaling algorithm and obtained a 2D representation of the distances between the users, as reported in Fig. 3. We notice that the distances between the users seem to reflect our qualitative observations.

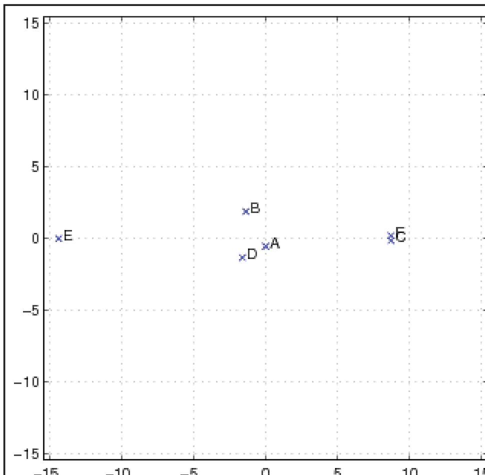

Fig. 3 2D representation of the distances between the users obtained by multidimensional scaling.

We then studied the possibility to change the whole movement with a single touch, instead of working with a movement as a sequence of postures.

More concretely we employed a Central Pattern Generator (CPG) to control the robot and showed that touch can be a powerful method to develop motions of CPG driven robots.

CPGs behavior depends on a huge number of parameters. Setting them by automatic search algorithms such as genetic algorithms (GA) has the drawbacks that the user has little control over the resulting motions, since it is often difficult to formulate criteria like human-likeness in terms of an evaluation function. Conversely hand-tuning is unintuitive, error-prone and time-consuming. We identified touch as a way to give the user full control over the resultant motion while keeping the user effort reduced.

In order to employ touch, predictability of the CPG behavior is fundamental; in fact we expect the user to require that similar touches lead to similar effects. We therefore designed a very predictable network of Hopf oscillators, and devised a protocol that maps touch patterns to CPG parameter changes, i.e. to changes in the movement.

In the experiments we used a simulator, which allows to easily discriminate between the touches due to gravity and the user touches, realized by mouse clicks in our system. Employing a simulator also permitted a direct comparison between setting the parameters by a genetic algorithm and setting them by touch. In particular we developed a crawling motion both by touching and by a genetic algorithm with evaluation function given by the average robot velocity in the direction of its head. Figures 4 and 5 provide a qualitative comparison of the two motions.

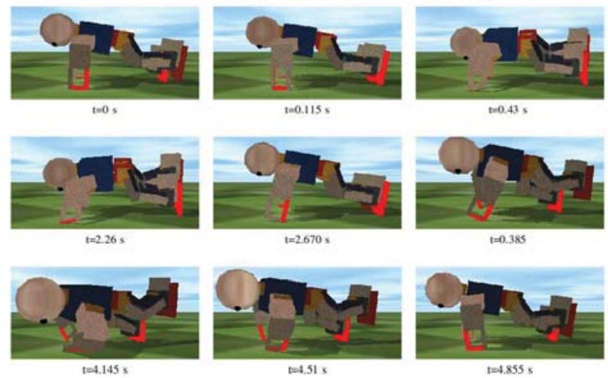

Fig. 4 Crawling motion developed using only touch interaction.

Fig. 5 Crawling motion developed using a genetic algorithm

Unless we carefully devise a very complicated evaluation function that takes care of many elements opportunely weighted it is difficult to obtain natural looking motions with an automatic parameter search, as can be easily seen in Fig. 5. In fact when humans develop motions by touching they implicitly consider a high number of criteria. This can often be seen afterwards analyzing the resulting motions. For instance the ranges of variation for the roll and the pitch of the robot are 15.4 and 14.5 degrees respectively for the motion obtained by tactile interaction and 34.7 and 23.9 for the motion optimized by the genetic algorithm.

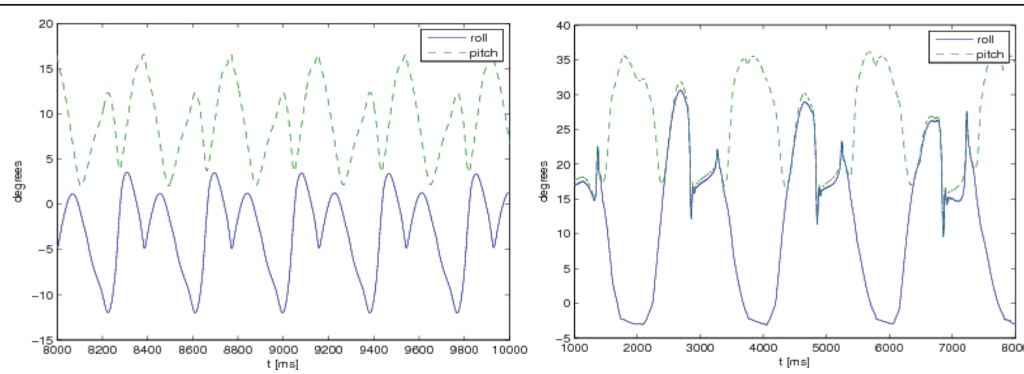

Fig. 5 Pitch and roll for the motion developed by touch (left) and for the one obtained by a GA (right).