Mapping between different kinematic structures without absolute positioning.

Erik Berglund*, Boyko Iliev*, Rainer Palm*, Robert Krug*, Krzysztof Charusta*, and Dimitar Dimitrov*
*Dept. of Natural Sciences and Technology
Örebro University, Sweden.
Email: firstname.lastname@oru.se

***Abstract*— When creating datasets for modelling of human grasping and manipulation skills based on training examples from human motion one can encounter the problem that the kinematics of the robot does not match the human kinematics. We present a simple method of bypassing the explicit modelling of the human kinematics.**

I. Introduction

In order to train a robot using human demonstrations one needs to measure the motion of the human hand, and translate this into commands for the robot. If there is an absolute reference for the motion of the human hand, such as a motion capture glove, the problem simply reduces to inverse kinematics, for which several adequate solutions exist. In other cases, when there is only indirect measurements of the positions of the human limbs, such as when using a CyberGlove [1], one needs to transform the sensor data from one domain into another (typically the domain of the joint-space of the robot being controlled), without actually knowing the transform. The problem considered in this paper is distinct from the task of a human controlling the robot directly, as in for example [2], where a CyberGlove is used as input device to demonstrate grasps in a virtual reality environment. In this case no precise mapping is required since the user is able to correct its own the hand pose in order to achieve the desired hand pose for the virtual hand. Other work has been done on Programming-by-demonstration, see for example [3] and [4], whereby a human 'teacher' creates a general pattern the robotic system can emulate. These works report that the grasp recognition rate is increased if absolute position sensors are used together with the CyberGlove.

One approach to this regression problem is to apply machine learning techniques. In this paper we apply the Parameter-Less Self-Organising Map 2 (PLSOM2) [5] algorithm.

In the absence of an absolute reference it becomes necessary to calibrate the model to the individual human 'teacher'. This is not a problem, as multiple examples by one teacher is sufficient, and additional teachers would only pollute the data set without adding to the solution. The problem we are looking at is mapping the input from a human thumb via a CyberGlove sensor glove to a ShadowHand [6] robotic thumb. We are only considering the motion of the thumb relative to the hand, not the motion of the hand through space. This paper is structured as follows: Section II describes the problem, section III discusses the basic approach we are presenting, section IV gives the details of how the system is trained, section V discusses the results, and section VI gives our conclusion.

Manuscript submitted 2011-03-01. This work was supported by the European Union as part of the HANDLE project.

II. Problem description

The problem of mapping the values from the CyberGlove to the ShadowHand is relatively straight-forward in the case of the index-, middle-, ring-, and little fingers. They have the same number of degrees of freedom as the corresponding human fingers, and the joints are of the same type and relative location. The only remaining unknown is the length of the links and the deflection of the human fingers. Even simply applying the joint values from the glove via simple offset and gain values (derived from calibration) is enough to get a tolerable accuracy. This is not the case for the thumb, as its structure is fundamentally different from the human thumb. The human thumb has four degrees of freedom (DOF): Two at the base and one each in the medial and distal joints. The ShadowHand thumb, by contrast, has five. To complicate matters, one of these DOFs are a rotational joint with its rotation axis parallel to the length axis of the thumb, see Figure 1. This is completely different from the human thumb, which has a saddle joint at the same location.

One way to solve this problem would be to use some kind of absolute position system to track the hand and the tip of the thumb in Cartesian space, thus allowing the joint angles to be solved using inverse kinematics. While this approach will in many cases be sufficient to provide accurate joint angles for the robot, it is not always applicable. Absolute position systems rely on carefully calibrated external sensors; the PhaseSpace [7] system relies on a set of at least three wall-mounted cameras, while the Polhemus [8] system requires a 'beacon' in proximity to the tracked object. If the subject moves outside the range of these sensors or the sensors are obscured in some fashion (magnetometric systems like Polhemus can be obscured by conducting materials), the accuracy quickly degrades. Furthermore these tracking systems require several transceivers to be placed on the hand, which reduces nimbleness and thus limits the type of grasps one can perform.

Another solution which would work without an absolute position system would be:

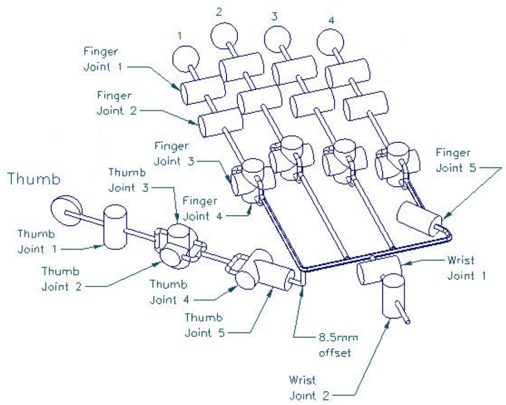

Fig. 1. Schematic view of ShadowHand kinematic structure. Note the number and alignment of joint in the thumb.

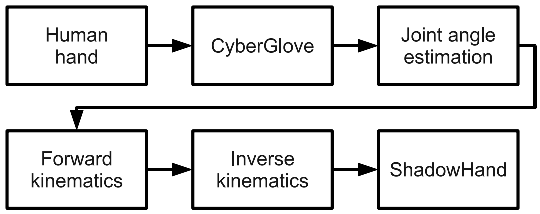

Fig. 2. A naive approach to mapping the human hand to the ShadowHand.

1) Read values from CyberGlove.
2) Estimate human joint angles.
3) Find the position of the human finger tip using forward kinematics.
4) Move the ShadowHand finger tip to the same location using inverse kinematics.

This procedure is outlined in Figure 2. This approach relies on having a kinematic model of the human hand. It also requires three discrete steps: Estimating the joint angles, modelling the kinematics of the human hand, and solving the inverse kinematics for the ShadowHand. Each of these steps are potential sources of errors.

Instead it might be desirable to bypass estimation of joint angles and forward and inverse kinematics by creating a direct model that maps from the values returned by the CyberGlove to joint angles for the ShadowHand, as outlined in Figure 3.

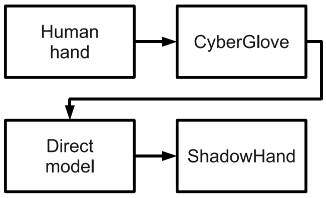

Fig. 3. A direct modelling approach.

One such approach was described in [3].

In the present approach a direct model is obtained by coupling ground truth readings of the fingertip position to corresponding values from the CyberGlove. Fingertip positions are collected using the PhaseSpace system. Based on the position of the fingertip relative to the rest of the hand, it is trivial to use inverse kinematics to find joint angles that let the ShadowHand achieve the same position.

III. Modelling approach

The Self-Organising Map (SOM) [9] family of algorithms (which includes the PLSOM2) can be seen as maps between real spaces of different dimensionality, see (1).

$$f : R^n \mapsto R^m, \tag{1}$$

where n is the dimensionality of the input space and m is the dimensionality of the output space. Typically, $n > m$, so that the map is dimensionality-reducing, but this is not a necessary condition. What characterises the SOM algorithms is that they preserve the topology of the input space [10]. Since there is reason to believe there is an underlying topology (that is, adjacent joint values are likely to map to similar finger positions), we select the PLSOM2 approach over other possible approaches such as, for example, the Back-Propagation Multi-Layer Perceptron.

The SOM is defined as a set of nodes, each associated with a prototype vector. When the map is presented with an input, the input is compared to the prototype vectors and the node with the prototype vector that most closely matches the input is the 'winning node'.

The PLSOM2 is an algorithm inspired by the SOM. The SOM requires the user to select at least three parameters before training:

- How quickly the learning rate should decrease.
- The initial value of the neighbourhood.
- How quickly the neighbourhood size should decrease.

The PLSOM2 simplifies this task by only requiring the user to select one parameter: The upper bound of the neighbourhood size. During training the neighbourhood size and the learning rate are calculated based on the training data. The PLSOM2 maintains an $n+1$ dimensional buffer of the inputs that are farthest from each other. This buffer is used to calculate an estimate of the diameter of the input space. The learning parameters are then calculated based on the distance from the current training input and the closest prototype vector relative to this diameter. If this distance is large, then the map has a poor representation of the current input and should make a large adjustment to improve. If this distance is small, then the map already has a relatively good representation of the current input and only a minor adjustment is necessary.

In the present case, $n = 4$, the number of values obtained from the CyberGlove for each sample. Also $m = 4$ in this case. The PLSOM2 contains a set of nodes embedded in the the m-dimensional output space. Each of these nodes are associated with a *label*, which is the centroid of all training examples that map to a given node after training is complete.

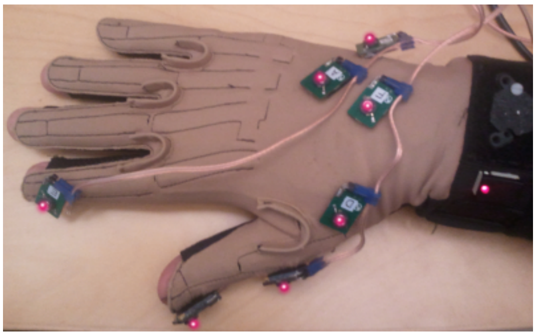

Fig. 4. The CyberGlove with attached PhaseSpace emitters.

In the present case the label corresponds to the joint angles of the ShadowHand thumb, and are thus 5-dimensional.

IV. Training the model

Data is collected using a right-handed 35-year old human male, a 21-DOF CyberGlove, and a PhaseSpace system with five cameras. The CyberGlove is equipped with three emitters to determine the position and orientation of the palm, and two emitters to determine the position of the tip of the thumb, see Figure 4. The raw data is available from [11]. The raw data is then processed by converting the PhaseSpace coordinates into joint angles for the ShadowHand using IK. The resulting data set is split into two using an 80%-20% split. The first 10,325 samples are the training set. The remaining 2,580 samples are the validation set and are not seen by the PLSOM2 during training. A PLSOM2 with 4096 nodes arranged in a 4-dimensional hypercube and a neighbourhood range of 6 is trained for 100,000 iterations using the training set. For each iteration one sample from the CyberGlove is drawn at random from the training set, then applied to the PLSOM2. On an entry-level desktop computer the training completes in less than 2 minutes.

At the end of the training, the map nodes are labelled with their corresponding joint angle values. If more than one joint angle maps to the same node, a centroid is calculated and used as the label. If no joint angles map to a node it is deleted.

V. Results

The validation set is presented to the PLSOM2 one sample at the time, and the label from the PLSOM2 (that is, joint angles) are applied to a kinematic model of the ShadowHand thumb, the finger tip position is calculated and compared to the ground truth from the recorded data. The mean distance between the desired location of the fingertip and the actual location is, on average, 19 mm.

At first glance this appears to be quite large, but should be seen in light of the inherent inaccuracy of the CyberGlove. Each joint monitored by the CyberGlove has a theoretical resolution of 256 discrete steps over the range of the joint. In reality only about half of the steps are actually used, which, at the tip of the thumb, gives an inherent error of circa 2 mm.

	Naive	PLSOM2
Mean error	44	19
Median error	46	17
Standard deviation	19	11

TABLE I
Positional error in millimetres.

Furthermore, there are also inherent inaccuracies stemming from problems of measuring the absolute position of the thumb and the hand. While the PhaseSpace system has a maximum resolution of 4 mm (determined by the size of the LEDs used as markers), they are mounted on the surface of the CyberGlove and therefore slide relative to the hand as the fingers are moved. Some of the diodes move by as much as 20 mm relative to the hand.

Finally, the carpometacarpal joint of a human thumb allows for up to 3 mm of distraction [12].

For comparison reasons we implemented the naive model described above using a transpose Jacobian IK solver. To provide maximum accuracy the algorithm was applied iteratively with steps of size 0.000025 mm until convergence. The model of the human hand had link lengths based on measurements of the subject's hand. The naive approach has as much as 44 mm average error on the same subset.

The results are described in Table I.

VI. Conclusion

By using a direct mapping between the CyberGlove and the ShadowHand we get the following advantages:

- Eliminates need for a kinematic model of human hand.
- No need for an absolute positioning system once training is complete.

Inverse kinematics must still be solved before training commences, and there is need for an absolute positioning system to gather the training data.

The inherent inaccuracy in this system demonstrates the need for intelligent controllers that can perform tasks such as maintain a grasp or in-hand manipulation by correctly interpreting the intentions of a human 'teacher'.

The proposed method achieves sufficient accuracy for qualitative analysis of motions and grasp recognition, though for reproduction of the exact fingertip position extra sensors are required.

References

[1] Cyberglove website. [Online]. Available: http://www.cyberglovesystems.com/

[2] J. Aleotti and S. Caselli, "Interactive teaching of task-oriented robot grasps," *Robotics and Autonomous Systems*, vol. 58, no. 5, 2010.

[3] R. Palm, B. Iliev, and B. Kadmiry, "Recognition of human grasps by time-clustering and fuzzy modeling," *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 484–495, 2009.

[4] K. Ikeuchi, K. Bernardin, K. Ogawara, and R. Dillman, "A sensor fusion approach for recognizing continuous human grasp sequences using hidden markov models." *IEEE Transactions on Robotics*, vol. 21, no. 1, 2005.

[5] E. Berglund, "Improved plsom algorithm," *Appl. Intell.*, vol. 32, no. 1, pp. 122–130, 2010.

[6] Shadow robot co. [Online]. Available: http://www.shadowrobot.com/

[7] Phasespace motion capture. [Online]. Available: http://www.phasespace.com/
[8] Polhemus motion tracking. [Online]. Available: http://www.polhemus.com/
[9] T. Kohonen, *Self-Organizing Maps*, 3rd ed. Springer, 2001.
[10] K. Kiviluoto, "Topology preservation in self-organizing maps," in *International Conference on Neural Networks*, 1996, pp. 294–299.
[11] Data used in generating the test data for this paper. [Online]. Available: http://pc124-178.oru.se/handle/icra2011data.txt
[12] G. R. Brunelli, "Stability of the first carpometacarpal joint," in *Finger bone and joint injuries*, A. Gilbert and P. Brüser, Eds. Taylor & Francis, 1999, pp. 167–170.