

## Analytical Upper Body Human Motion Transfer to Naohumanoid Robot

Adrianto Ravi Ibrahim and Widyawardana Adiprawita

School of Electrical Engineering and Informatics, ITB

**Abstract:** This paper proposes an analytical method to solve the problem of upper body motion transfer from human to humanoid robot. By using humanoid normalized model as a model of upper body human motion, the proposed method then uses analytical inverse kinematics method to transfer human motion to humanoid robot. Because the nature of analytical method that is specific to robot with particular kinematics structure, we choose the Aldebaran NAO robot as target platform. We verify our method by using offline simulation and online experiment in the robot. The offline simulation shows that our method gives exact solution, while the online experiment gives a qualitatively good result.

### 1. Introduction

Recently, a new vision sensor that allows the tracking of human motion without marker becomes available worldwide for a relatively cheap value. In the context of robotics, this sensor opens up the possibility of vision-based imitation learning or gesture-based humanoid teleoperation. Given a data or information of human pose at a certain time, both imitation learning and gesture-based teleoperation share a common problem, namely the motion transfer problem.

Motion transfer problem is a problem of “translating” motion/pose of an object to another object. An example of motion transfer problem is the problem of making humanoid robot imitates human movement sampled by motion-tracking sensor (Figure 1).

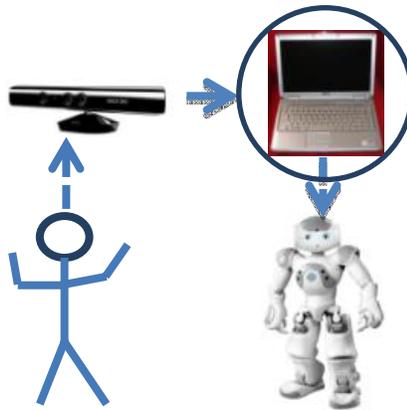


Figure 1. A Gesture-Based Teleoperation System.  
A human (in blue) moves in front of a motion tracking sensor.  
The robot then imitates the movement.

In Figure 1, a human moves in front of a sensor that can detect human movement. This movement data is then “translated” and sent to humanoid robot. The robot moves according to the “translated” data. If the “translation” is done correctly, then the robot movement will be “similar” to human movement.

Although many have addressed this motion transfer problem, the method used is based either on optimization (i.e. differential kinematic [1-3]) or machine learning (i.e. Gaussian process [4]). Although these two methods are appropriate for complex motion transfer such as whole body motion, we believe that a simple motion transfer problem such as arms movement can be solved by using a simpler analytical method. In this paper, we show that it is possible to solve human arms motion transfer problem analytically with the assumption that the robot joints angle is in the range  $[-\pi, \pi]$  rad:

- We introduce the notion of generalized kinematic to deal with motion transfer problem (Section III);
- We devote the entirety of Section IV to give a detailed solution for upper body motion transfer problem to NAO;
- We do an offline simulation and direct experiment to measure the quality of our method (Section V)

## 2. System Overview

Our motion transfer system contains three main elements (Figure 2): motion tracking sensor Kinect from Microsoft, humanoid robot NAO from Aldebaran Robotics, and a robotic framework ROS (Robot Operating System).

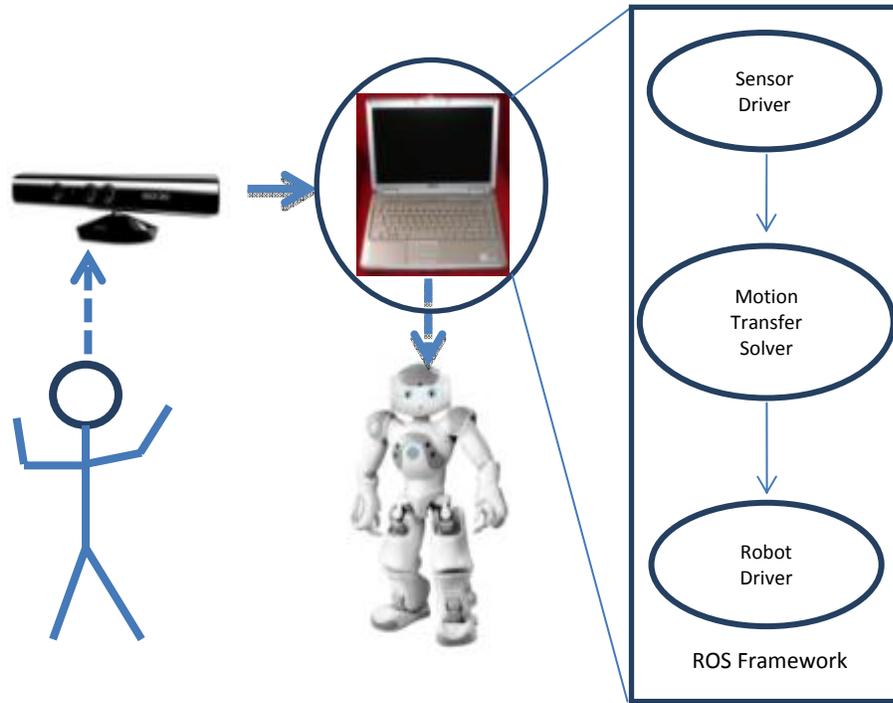


Figure 2. Motion Transfer System

The main elements of our motion transfer system (sensor, robot, and ROS framework) The ROS framework glues together various programs we developed: sensor driver, motion transfer solver, and humanoid robot driver.

### A. Sensor

We use Kinect (Figure 3) as motion tracking sensor in our motion transfer system. This sensor captures depth information as well as color information of a scene. For capturing motion data, we use library provided by OpenNI [6] and Primesense [7].



Figure 3. Microsoft Kinect

### B. The Humanoid Robot NAO

NAO is a humanoid robot created by the French based Aldebaran Robotics company. This robot has 5 kg weight and 58 cm height. NAO has 23 degrees of freedom: 2 degrees of freedom head, 4 degrees of freedom for each arm, 1 degree of freedom for pelvis, and 5 degrees of freedom for each leg. From [8], coordinate system when all joints angle are at 0 rad is shown in Figure 4.

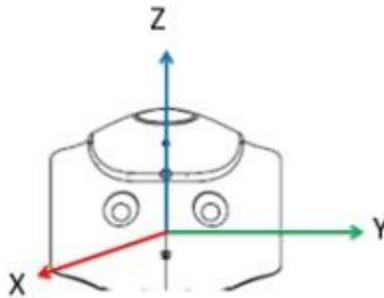


Figure 4. Reference Coordinate System

There are two link descriptions for NAO. The first description uses Denavit-Hartenberg parameters [8], while the second description uses rotation axis and offset of a joint. For convenience in deriving generalized kinematic for NAO, we use the second description in this paper (Table 1).

Table 1. NAO Arm Joints Parameters

Joint Name	Rotation Axis ( $x_i$ )	Offset ( ${}^{i-1}O$ )
LShoulderPitch	$y$	(0, 98, 0)
LShoulderRoll	$z$	(0, 0, 0)
LElbowYaw	$x$	(90, 0, 0)
LElbowRoll	$z$	(0, 0, 0)
LWristYaw	$x$	(108.5, 0, 0)
RShoulderPitch	$y$	(0, -98, 0)
RShoulderRoll	$z$	(0, 0, 0)
RElbowYaw	$x$	(90, 0, 0)
RElbowRoll	$z$	(0, 0, 0)
RWristYaw	$x$	(108.5, 0, 0)

C. *Robotic Operating System (ROS)*

ROS is a robotic framework designed to ease the development of robotic software. The central idea of ROS is the concept of node and topic. A node is a computational unit. A node can be a hardware driver, an algorithm, or a controller for a robot.

Two nodes communicate with each other by using messages. There are two models of messages passing supported in ROS. The first one is a service based message passing method. In this model, a node requests a service from another node by sending a request message to the service providing node. The service providing node will then answer the message by sending the result to the node that requested the service earlier. This model is basically a one-to-one communication between nodes.

The second model is the publisher-subscriber model of communication. In this model, publisher node publishes its message to a certain topic. Any nodes that need the message from the publisher node subscribes to the topic. We use this second model in our motion transfer system. We implemented our motion transfer system as three nodes: the sensor node, the solver node, and the controller node. The sensor node is a wrapper for sensor driver provided by OpenNI [6] and Primesense[7]. The solver node solves the motion transfer problem, while the controller node controls NAO remotely.

**3. Kinematics**

In typical analytical inverse kinematic framework (i.e. industrial manipulator), kinematic constraints are defined in terms of target end-effector pose. For example, suppose end-effector *A* in Figure 5 is at a position  $(x_1, y_1)$  and the target pose *C* is at position  $(x_2, y_2)$ .

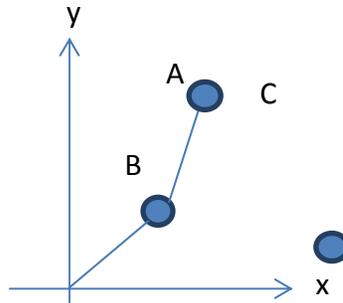


Figure 5. Typical Setup in Analytical Inverse Kinematic

In this kind of setup, we usually ask two questions: whether *A* can reach *C*, and if *A* can reach *C*, what the configuration of joint *A* and joint *B* is. For a redundant manipulator, both questions can be answered using analytical inverse kinematic method [11-13]. The setup in motion transfer problem is different though. Suppose there are two different arms *G* and *H* (Figure 6).

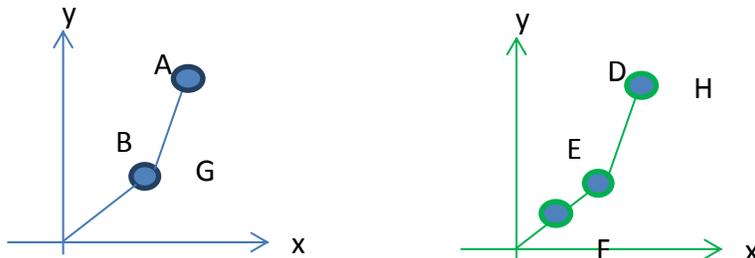


Figure 6. Setup for Motion Transfer Problem

The question we would like to ask in this setup is how arm  $H$  must be configured such that arm  $H$  “imitates” arm  $G$ . The notion “imitates” in this example means that there must be some “similarity” between movement of arm  $G$  and movement of arm  $H$ . Intuitively, the movement of arm  $H$  must at the very least look “similar” to the movement of arm  $G$ . This means that we need a set of constraints that cover all joints in arm  $H$  instead of constraint that is only related to end-effector joint. One possible set of constraints for this example is a set containing constraints in the form of two vectors describing two segments in arm  $H$ . We’d like to note that these constraints are not directly related to the end-effector. We call these kinds of constraints as generalized kinematic constraints.

Specifics to our problem, human upper body motion transfer to NAO, the kinematic constraints we chose are inspired by the works of Kulpa et al. [10] in computer graphics and the optimization based motion transfer [1-3]. We choose to describe human movement as two unit vectors: A unit vector in the direction of upper arm; and a unit vector in the direction of lower arm.

#### 4. Motion Transfer

##### A. Notation

Throughout this section, we will use notations defined here:

$\Sigma_i$  is a coordinate system;  $\Sigma_0$  is the reference coordinate system;

$\mathbf{A} = (\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3)$  is a vector;

$\mathbf{O}_s$  is origin of coordinate system  $\Sigma_s$  under  $\Sigma_0$ ;

$\mathbf{R}_a(\theta)$  is a  $\theta$  rad rotation around axis  $\mathbf{a}$  in current coordinate system;

$\mathbf{U}(d_{ab})$  is a translation in the direction of current  $x$  axis with magnitude  $d_{ab}$ ;

${}^A_B\mathbf{T}$  is a homogeneous transform matrix from coordinate system  $\Sigma_A$  to coordinate system  $\Sigma_B$ ;

##### B. Kinematic Model of Manipulator

A simple manipulator with  $n$  degrees of freedom can be described as a kinematic chain with  $n$  joints. In this paper, we define a manipulator as follows:

1. A manipulator of  $n$  degree of freedoms consists of  $n + 1$  coordinate systems;
2. The reference coordinate system  $\Sigma_0$  coincides with the first joint in the kinematic chain;
3. The origin of coordinate systems  $\Sigma_i (i = 1, 2, 3, \dots, n)$  is located at each joint position  $\mathbf{p}_i$  in the reference coordinate system  $\Sigma_0$ ;
4. The axes of each coordinate systems  $\Sigma_i (i = 1, 2, 3, \dots, n)$  parallel to the axes of reference coordinate system  $\Sigma_0$  when all joint angles  $\boldsymbol{\theta} = \mathbf{0}$ .

With this description of manipulator, a coordinate system in this manipulator can be characterized by two parameters which are its rotation axis  $\mathbf{x}_i$ , and its origin with respect to the previous coordinate system  ${}^{i-1}_i\mathbf{O}$ .

##### C. Generalized Kinematic Constraints for Human Arm

We define generalized kinematic constraints for human arm in terms of two unit vectors. The first vector  $\mathbf{U}$  is a unit vector with the same direction as the direction of upper arm in human torso coordinate system  $\Sigma_M$ . The second vector  $\mathbf{L}$  is a unit vector in the direction of lower arm also in human torso coordinate system  $\Sigma_M$ .

Given a set of points  $A = \{\mathbf{x}_s, \mathbf{x}_e, \mathbf{x}_h\}$  containing the position of shoulder ( $\mathbf{x}_s$ ), elbow ( $\mathbf{x}_e$ ) and hand ( $\mathbf{x}_h$ ) in the sensor’s coordinate system  $\Sigma_K$  and homogeneous transformation matrix  ${}^M_K\mathbf{T}$  from torso coordinate system  $\Sigma_M$  to the sensor’s coordinate system  $\Sigma_K$ , define

$$\mathbf{U} = \frac{{}^M_K\mathbf{T} \frac{\mathbf{x}_e - \mathbf{x}_s}{\|\mathbf{x}_e - \mathbf{x}_s\|}}{\| \frac{{}^M_K\mathbf{T} \frac{\mathbf{x}_e - \mathbf{x}_s}{\|\mathbf{x}_e - \mathbf{x}_s\|} \|} \quad (1)$$

$$\mathbf{L} = \frac{{}^M_K\mathbf{T} \frac{\mathbf{x}_h - \mathbf{x}_e}{\|\mathbf{x}_h - \mathbf{x}_e\|}}{\| \frac{{}^M_K\mathbf{T} \frac{\mathbf{x}_h - \mathbf{x}_e}{\|\mathbf{x}_h - \mathbf{x}_e\|} \|} \quad (2)$$

We use(1) and (2) as the constraints in solving the motion transfer problem.

#### D. Kinematic Model of NAO Robot's Arm

In the robot's end, there are two equations needed to be specified in our method. The first relates NAO's upper arm unit vector  $\mathbf{U}_{NAO}(\boldsymbol{\theta})$  in Cartesian space and NAO's arm joints angle. The second equation relates NAO's lower arm unit vector  $\mathbf{L}_{NAO}(\boldsymbol{\theta})$  and NAO's arm joints angle.

To define these two equations, we define four coordinate systems located on the body of NAO. The first coordinate system  $\Sigma_N$  is located on the torso joint of NAO. This coordinate system serves as reference for the other coordinate systems. The second coordinate system  $\Sigma_s$  is located on the shoulder joint of NAO. The origin  $\mathbf{O}_s$  of this coordinate system is

$$\mathbf{O}_s(\boldsymbol{\theta}) = \mathbf{l}(d_{ts}) \quad (3)$$

The third coordinate system  $\Sigma_E$  is located on the elbow joint of NAO. The origin  $\mathbf{O}_e$  of this coordinate system is

$$\mathbf{O}_e(\boldsymbol{\theta}) = \mathbf{l}(d_{ts}) + \mathbf{R}_y(\theta_1)\mathbf{R}_z(\theta_2)\mathbf{l}(d_{se}) \quad (4)$$

The mapping from the joint space of NAO robot to the upper arm vector  $\mathbf{U}$  is defined in terms of (3) and (4):

$$\mathbf{U}_{NAO}(\boldsymbol{\theta}) = \frac{\mathbf{O}_e(\boldsymbol{\theta}) - \mathbf{O}_s(\boldsymbol{\theta})}{d_{se}} = \frac{\mathbf{R}_y(\theta_1)\mathbf{R}_z(\theta_2)\mathbf{l}(d_{se})}{d_{se}} \quad (5)$$

To define the second mapping,  $\mathbf{L}_{NAO}(\boldsymbol{\theta})$ , we need to define the hand coordinate system  $\Sigma_H$  first. Define hand coordinate system  $\Sigma_H$  to be located in the hand joint of NAO. The origin of this coordinate system  $\mathbf{O}_h$  is given by

$$\mathbf{O}_h(\boldsymbol{\theta}) = \mathbf{l}(d_{ts}) + \mathbf{R}_y(\theta_1)\mathbf{R}_z(\theta_2)\mathbf{l}(d_{se}) + \mathbf{R}_y(\theta_1)\mathbf{R}_z(\theta_2)\mathbf{R}_x(\theta_3)\mathbf{R}_z(\theta_4)\mathbf{l}(d_{eh}) \quad (6)$$

With the coordinate system  $\Sigma_H$  defined, the mapping  $\mathbf{L}_{NAO}(\boldsymbol{\theta})$  can now be defined as follows

$$\mathbf{L}_{NAO}(\boldsymbol{\theta}) = \frac{\mathbf{O}_h(\boldsymbol{\theta}) - \mathbf{O}_e(\boldsymbol{\theta})}{d_{eh}} = \frac{\mathbf{R}_y(\theta_1)\mathbf{R}_z(\theta_2)\mathbf{R}_x(\theta_3)\mathbf{R}_z(\theta_4)\mathbf{l}(d_{eh})}{d_{eh}} \quad (7)$$

#### C Solving Generalized Kinematic of NAO's Arms

After we define  $\mathbf{U}$ ,  $\mathbf{L}$ ,  $\mathbf{U}_{NAO}$ , and  $\mathbf{L}_{NAO}$ , we can reduce the problem of upper body human motion transfer to humanoid to solving the equations

$$\mathbf{U}_{NAO}(\boldsymbol{\theta}) = {}^N_K\mathbf{T}\mathbf{U} \quad (8)$$

$$\mathbf{L}_{NAO}(\boldsymbol{\theta}) = {}^N_K\mathbf{T}\mathbf{L} \quad (9)$$

where  ${}^N_K\mathbf{T}$  is a homogeneous transform matrix from NAO's torso coordinate system  $\Sigma_N$  to the sensor's coordinate system  $\Sigma_K$ . Generally, solution to (8) and (9) can be divided into two kinds of solution, solution under a non-singular configuration and solution for singular configurations.

The solution for non-singular configuration can be found by directly solving (8) and (9). Setting  $\mathbf{U}' = {}^N_K\mathbf{T}\mathbf{U}$  for the first equation, we can determine the value of  $\theta_1$  and  $\theta_2$ :

$$\theta_1 = \tan^{-1}(-u'_3, u'_1) \quad (10)$$

$$\theta_2 = \tan^{-1}(-u'_2, \sqrt{u'_1{}^2 + u'_3{}^2}) \quad (11)$$

Once the value of  $\theta_1$  and  $\theta_2$  are found, we can also determine the value of  $\theta_3$  and  $\theta_4$ . Let  $\mathbf{L}' = [\mathbf{R}_y(\theta_1)\mathbf{R}_z(\theta_2)]^{-1} \mathbf{N}_K^T \mathbf{T}$ . The solution for NAO's right arm is given in (12) and (13), while the solution for NAO's left arm is given by (14) and (15).

$$\theta_3 = \tan^{-1}(l'_3, l'_2) \quad (12)$$

$$\theta_4 = \tan^{-1}(\sqrt{l'_2{}^2 + l'_3{}^2}, l'_1) \quad (13)$$

$$\theta_3 = \tan^{-1}(-l'_3, -l'_2) \quad (14)$$

$$\theta_4 = -\tan^{-1}(\sqrt{l'_2{}^2 + l'_3{}^2}, l'_1) \quad (15)$$

For singular configuration, we must consider three cases:

1.  $\theta_2 = \frac{\pi}{2}$ ; When this happens, the first joint in chain will lose its degree of freedom. As such, the value of  $\theta_1$  could not be determined from  $\mathbf{U}$ . However, because changing the value of  $\theta_1$  will not change the position of elbow joint,  $\theta_1$  could be set to 0. Setting  $\mathbf{L}' = \mathbf{N}_K^T \mathbf{T}$  as above, we can determine the value of  $\theta_3$  and  $\theta_4$ :

$$\theta_3 = \tan^{-1}(l'_3, -l'_1) \quad (16)$$

$$\theta_4 = \tan^{-1}(\sqrt{l'_1{}^2 + l'_3{}^2}, l'_2) \quad (17)$$

2.  $\theta_2 = -\frac{\pi}{2}$ ; Using the same method as the first case, we can determine the value of  $\theta_3$  and  $\theta_4$ :

$$\theta_3 = \tan^{-1}(l'_3, l'_1) \quad (18)$$

$$\theta_4 = \tan^{-1}(\sqrt{l'_1{}^2 + l'_3{}^2}, -l'_2) \quad (19)$$

3.  $\theta_4 = 0$ ; When this happens, the lower arm vector is essentially pointing to the same direction as the upper arm vector. Thus, we can find joint space configuration by simply solving the equation of upper arm vector. The value of  $\theta_3$  and  $\theta_4$  can be set to arbitrary value because these values will not change the position of the hand joint. For simplicity, we set these values to 0 rad.

## 5. RESULT AND DISCUSSION

### A. Result

We test our method by using offline simulation and direct testing in the robots. In offline simulation, we capture ten datasets of pose using the sensor Kinect. Five datasets correspond to dynamic pose or movement (Figure 7), while the other five datasets consist of still (Figure 8) pose. Each dataset contains 100 samples taken at sampling frequency 30 Hz.

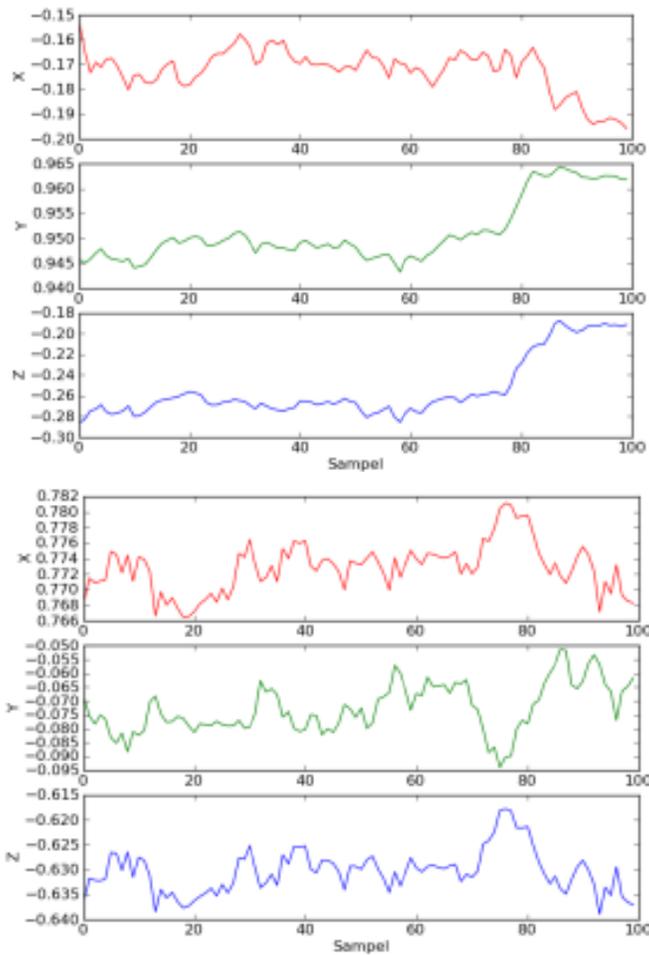


Figure 7 Samples of Still Poses Trajectory (vertical axis is in meter)

Error measure for our method is defined not for the joint angles of the robot but for the direction of the related limb of the robot. In other words, the measurement is not done directly on the joint angles, but indirectly by measuring on the direction of the upper arm and the lower arm of NAO robot. Formally, let  $\vec{u}$  the solution given by our method with input upper arm vector  $\vec{u}_i$  and lower arm vector  $\vec{l}_i$ . The error measure for the vector pairs  $(\vec{u}_i, \vec{l}_i)$  and  $(\vec{u}, \vec{l})$  is defined as follows

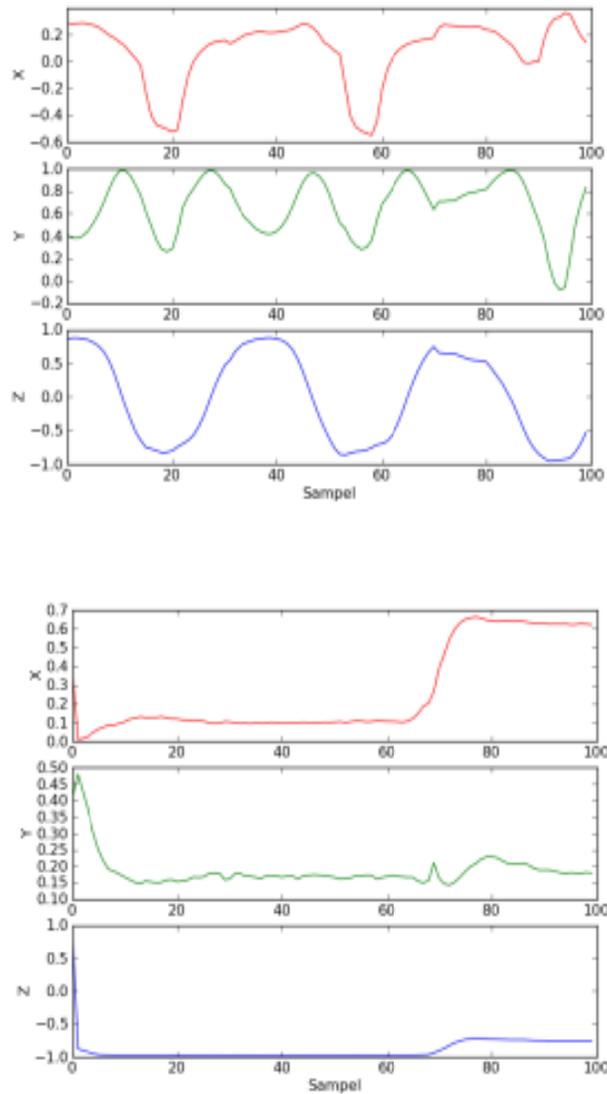


Figure 8 Samples of Dynamic Poses Trajectory (vertical axis in meter)

Using  $\theta$  and  $\phi$  as error measure, we found that for all samples, both  $\theta$  and  $\phi$  equals 0 rad. This means that our method gives exact solution for this motion transfer problem.

In the online test, we use qualitative measure only. In this test, we ask a student to stand in front of the sensor, and to move her arm (Figure 9).



Figure 9. Three Samples of Online Test Poses

From Figure 9, we can see that NAO's poses "look" the same as the student poses.

### B. Discussion

Although our method gives good solution, there are still many practical problems that need to be solved before this method can be used reliably for imitation learning or gesture-based teleoperation. We observe some problems that need to be resolved before this method can be of practical use. The first problem is the problem of joint limit. Because of the limitation of NAO's joints, there are many poses that could not be imitated by the robot. Analytical solution for this problem was given by [11], [12], but unfortunately, because of NAO's structure, this we could not use this method for NAO. The solution in [11], [12] assumes the existence of redundant circle in elbow joint that NAO does not have. A possible solution for this problem then is by using optimization based kinematic for avoiding joint limitation as in [2], [3].

Another problem is the problem of correspondence between the robot and human user. The robot needs time to move its actuator from a position to another position. Consequently, at time  $t$ , it is possible that the robot is still executing pose from time  $t - \tau$ . This becomes a problem because the human user must then wait until the robot finish executing current pose before moving to the next pose. This is impractical in the context of gesture-based teleoperation. A predictive filter might be a possible solution for this problem.

## 6. Related Works

### A. Machine Learning Based Solution

This method was proposed by Shon et al. [4] and used Gaussian process to reduce the dimension of human pose's space and robot pose's space to a common latent space. This method interprets the problem of motion transfer as a nonlinear regression problem from both the human pose's space and robot pose's space to the common latent space. In the learning stage of this algorithm, the algorithm learns the common latent space by using data of human pose and robot pose. When the algorithm encounters new human pose, it then maps the pose in the latent space and finds the inverse mapping from the latent space to the robot pose's space. Although this work is very interesting because it does not rely on the underlying kinematic structure at all, it does not align with our goal to investigate the possibility of using analytical method for solving motion transfer problem.

### B. Differential Kinematic

Differential kinematic is a method to solve inverse kinematic problem by iteratively finding a possible solution using the Jacobian of the generalized kinematic equation. The use of differential kinematic for solving motion transfer problem was proposed the first time by Choi and Ko [1] for solving motion transfer in the field of computer graphics. Recently, this method has been successfully employed in the field of robotics by Dariush et al. [2] and Montecillno et al. [3] for whole body humanoid motion transfer. We are inspired by the notion of generalized kinematic in [3] to solve motion transfer problem.

### C. Analytical Inverse Kinematic

Only in a few month after the first release of Kinect sensor, Taylor Veltrop [5] successfully made humanoid robot imitates his movement remotely. Although we use the same platform as Taylor Veltrop (ROS with NAO), our implementation is very different though. Taylor Veltrop used a geometrical approach in solving the motion transfer problem. He used triangles and law of cosines to find all related angles for a movement. In the other hand, we use a more algebraic approach. We started by specifying a model of human arms, a model of NAO's arms, and then we solve equations relating those two models. As a result, our approach is more explicit about the constraints and we pose on the robot. Besides that, by using algebraic approach, we can deal with singularity configuration that is hard to detect if we use geometrical approach.

## 7. Conclusion

This paper provides a preliminary work for solving motion transfer problem using analytical method. In simulation, the method proposed in this paper gives exact solution under the assumption that joint limit is not present. When testing directly in the robot, the robot's poses look the same as human's poses. For the future work, to improve the performance of the proposed method, analysis of joint limit and the correspondence problem will be taken into consideration for the motion transfer problem.

## References

- [1] K. J. Choi and H. S. Ko, "On-line motion retargeting," *Journal of Visualization and Computer Animation*, vol. 11, pp. 223–235, 1999.
- [2] B. Dariush, M. Gienger, A. Arumbakkam, C. Goerick, Y. Zhu, and K. Fujimura, "Online and markerless motion retargeting with kinematic constraints," in *Proceedings of the IEEE/RAS Intelligent Robots and System Conference (IROS)*, 2008.
- [3] F. J. Montecillo-Puente, M. N. Sreenivasa, and J.P. Laumond, "On real-time whole-body human-humanoid motion transfer," in *Proceedings of the International Conference on Informatics and Control, Automation, and Robotics (ICINCO)*, 2010.
- [4] A. P. Shon, K. Grochow, and R. P. N. Rao, "Robotic imitation from human motion capture using gaussian processes," in *In Proceedings of the IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, 2005.

- [5] T. Veltrop, <http://www.ros.org/wiki/openni/Contests/ROS3D/HumanoidTeleoperation>.
- [6] <http://openni.org/>.
- [7] <http://www.primesense.com/>.
- [8] [http://users.aldebaran-robotics.com/docs/site\\_en/reddoc/index.html](http://users.aldebaran-robotics.com/docs/site_en/reddoc/index.html).
- [9] D. Gouaillier, V. Hugel, P. Blazevic, C. Kilner, J. Monceaux, P. Lafourcade, B. Marnier, J. Serre, and B. Maisonnier, "The nao humanoid: a combination of performance and affordability," *CoRR*, July, 2008.
- [10] R. Kulpa, F. Multon, and B. Araldi, "Morphology-independent representation of motions for interactive human-like animation," in *Eurographics 2005*, vol. 24, no. 3, 2005.
- [11] P. Dahm and F. Joubin, "Closed form solution for the inverse kinematics of a redundant robot arm," *Inst. Neuroinf, Ruhr-Univ. Bochum, 44780, Bochum, Germany, Internal Report 97-08*, April 1997.
- [12] M. Shimizu, H. Kakuya, W.-K. Yoon, K. Kitagaki, and K. Kosuge, "Analytical inverse kinematic computation for 7-dof redundant manipulators with joint limits and its application to redundancy resolution," *IEEE Transactions on Robotics*, pp. 1131–1142, 2008.
- [13] L. M. Miller, H. Kim, and J. Rosen, "Redundancy and joint limits of a seven degree of freedom upper limb exoskeleton," in *Proceedings of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2011.
- [14] B. Dariush, Y. Zhu, A. Arumbakkam, and K. Fujimura, "Constrained closed loop inverse kinematics," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2010.
- [15] O. Kanoun, "Real-time prioritized kinematic control under inequality constraints for redundant manipulators," in *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2011.



**Adrianto Ravi Ibrahim** graduated from his Bachelor degree program in Electrical Engineering at Bandung Institute of Technology (ITB), Indonesia, in April 2012. He was a student exchange at University of Electro-Communication, Tokyo, Japan from October 2010 to September 2011. His interests include robotics, control system, and signal processing.



**Widyardana Adiprawita** lecturer at STEI-ITB. Received his electrical engineering degree at Electrical Engineering ITB with honor in 1997. Finished master degree at Informatics Engineering ITB in 2000. And finished doctoral degree with honor at Electrical Engineering ITB. His Research interests are embedded system, robotics and intelligent agent autonomy. He has written more than 20 papers published in international and national publication.