2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)
July 6-11, 2014, Beijing, China

Fuzzy-based Adaptive Motion Control of a Virtual iCub Robot in Human-Robot-Interaction

Zejun Xu, Chenguang Yang*, Hongbin Ma*, Menyin Fu

***Abstract*— In this paper, in order to combine intelligence of human operator and automatic function of the robot, we design a control scheme for the bimanual robot manipulation, in which the leading robot arm is directly manipulated by a human operator through a haptic device and the following robot arm will automatically adjust its motion to match the operator's motion. In this paper, we propose a fuzzy-based adaptive feed-forward compensation controller and apply it into the robot control. According to the comparison results in the simulated experiment, we conclude that the fuzzy-adaptive controller performs better than the non-fuzzy controller, although they can both complete the specified task by tracking the leading robot arm controlled by the human operator. The techniques developed in this paper could be very useful for our future study on adaptation in human-robot interaction in improving the reliability, safety and intelligence.**

I. Introduction

In the past few decades, robotics has developed rapidly. The first generation robots, which mainly track the pre-designed trajectory, are mainly applied in industry. The second generation of robots, which are partially empowered by computer vision technology, are able to sense the change of external environments and adjust its behavior adaptively. The third generation intelligent robots under development are able to complete the task independently[1]. From the trend above, we see that robots are being developed towards the direction of multifunction and intelligence.

Recently, more and more scholars have paid attention to the research in humanoid robots. The iCub robot, as is shown in Fig.1, is a humanoid robot developed as part of the EU project RobotCub[2] and subsequently adopted by more than 20 laboratories worldwide. It has 53 motors that move the head, arms, hands, waist, and legs. Some experiments have been made on this robot. In [3], the iCub robot is controlled to draw shapes after observing a demonstration by a teacher, by using a series of self evaluations of its performance. In [4], a technique is proposed for controlling the interaction forces exploiting a proximal six axes force/torque sensor. This technique is tested and validated on the iCub robot. One advantage of iCub robot is that it is an open source cognitive robotic platform.

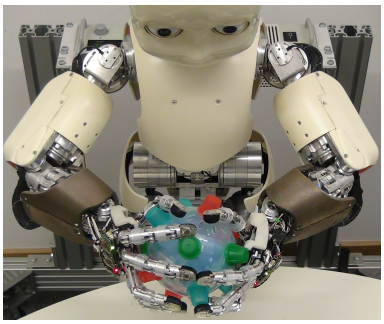


Fig. 1: The iCub Robot *[Photo taken in Plymouth University]*

Much research has been performed on intelligent robots which have autonomous learning abilities. In [5], an adaptive control scheme of coordinated motion of spacecraft attitude and the desired trajectory in the manipulator's workspace is proposed. A behavior-based intelligent controller is designed for the control of a 2-DoF (degree of freedom) manipulator in [6] which can learn the behavior online and does not rely on the system model. In [7], a Q-learning control algorithm is applied to the control of robot manipulator. In [8], a model based on disturbance attenuator (MBDA) is put forward and the manipulator is controlled using MBDA, which resolves the problems caused by an unknown and uncertain mathematical model. Although scholars have made much effort on the control problem of robot manipulators and have proposed numerous methods to overcome the difficulty such as uncertainties of the robot dynamic model and nonlinearity in robot, most of the methods are subject to some limitations. That means robots can only complete some fixed repeated and easy tasks. Due to the limitation of artificial intelligent technique, we may not obtain the satisfied performance if we totally rely on robot intelligence to complete complex tasks.

As it is unfeasible for us to totally rely on robots, so we desire to combine the human operator's intelligence and the techniques of robots autonomous motion control. In this paper, we introduce human operator's guidance into the system such that the robot could understand human operator's behavior and will adjust its own motion state adaptively according to human operator's guidance.

In this paper, we employ the virtual iCub robot as the research platform to study human robot interaction (HRI).

This work was supported in part by EU Grant PIIFR-GA-2010-910078 , the Royal Society research grant RG130244; the National Natural Science Foundation in China (NSFC) under Grants 61004059 and 61203074, NSFC-RS Joint Project grant (61211130359 and IE111175), and Beijing Outstanding Talents Programme (2012D009011000003).

* To whom all correspondences should be addressed. Tel: (+86)-10-68913985, E-mail: mathmhb@bit.edu.cn;cyang@ieee.org

The authors are with School of Automation as well as State Key Laboratory of Intelligent Control and Decision of Complex Systems, Beijing Institute of Technology, Beijing, China, 100081; Chenguang Yang is also with School of Computing and Mathematics, Plymouth University , UK, PL4 8AA.

978-1-4799-2072-3/14/$31.00 ©2014 IEEE

II. Problem Formulation

As is shown in Fig.2, the task of the iCub robot is to move an object along the trajectory specified by a human operator. The leading manipulator of iCub robot is manipulated directly by position command set by human operator through Falcon joystick.

The object between the two hands of the robot is supposed to move synchronously with the leading manipulator. Equivalently, there is no relative motion between the object and the leading manipulator. The object dynamics is modeled as a combination of spring-mass-dampers, which is similar to that in [9], and the effect of gravity is not in consideration.

The following manipulator will move autonomously under the designed controller. Because the following manipulator of iCub robot is in contact with the object, when the object is moving, the force caused by deformation will affect the motion of the former. The force between the following manipulator and the object will be detected and added into the control law of the designed controller.

Once the distance between two hands of iCub robot is too large, the object will fall down on the ground. On the other hand, if the distance is too small, the object will be damaged. Ideally, the following manipulator could track the trajectory of the leading one while the distance between both hands maintains a constant, so the object can be carried naturally and smoothly. In order to simplify the process, we just consider the motion in one dimension. Table I lists the main notations used in this paper.

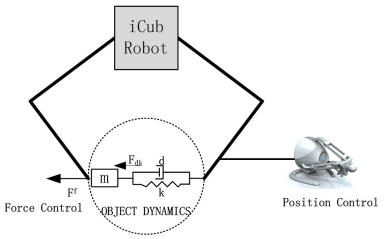


Fig. 2: Illustration of object dynamics and control scheme

TABLE I: Nomenclature

Symbol	Description
q^f, q^l	Joint angles of the following and the leading hand, respectively
q_d^f	Desired joint angles of the following hand
X^f, X^l	Actual Cartesian position of the following and the leading hand
X_d^f	Desired Cartesian position of the following hand
τ	Joint torque
F	Force in Cartesian space
e	Position error
d	Damping coefficient
k	Elasticity coefficient
r	Radius of the object (ball)
p	Desirable distance between two hands

TABLE II: D-H parameters of an arm of iCub robot

Link i	$a_i(mm)$	$d_{i+1}(mm)$	$\alpha_i(rad)$	$\theta_{i+1}(deg)$
$i=0$	0	107.74	$-\frac{\pi}{2}$	$90+(5\rightarrow 95)$
$i=1$	0	0	$\frac{\pi}{2}$	$-90+(0\rightarrow 161)$
$i=2$	15	152.28	$\frac{-\pi}{2}$	$75+(-37\rightarrow 100)$
$i=3$	-15	0	$\frac{\pi}{2}$	$5.5\rightarrow 106$
$i=4$	0	137.3	$\frac{\pi}{2}$	$-90+(-50\rightarrow 50)$
$i=5$	0	0	$\frac{\pi}{2}$	$90+(10\rightarrow -65)$
$i=6$	62.5	-16	0	$-25\rightarrow 25$

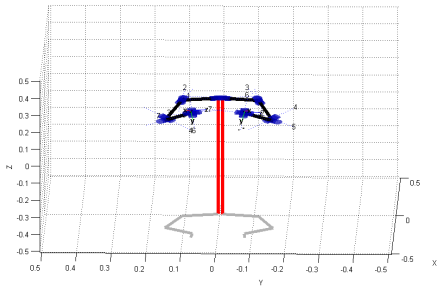


Fig. 3: Simulated iCub robot (Using MATLAB RVC Toolbox[13])

A. Dynamics of Robot Manipulator

The dynamics of a single robot manipulator is set by the following equation:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + \tau_{ext} + G(q) = \tau \tag{1}$$

where q denotes the joint position of the robot arm, $M(q) \in \mathbb{R}^{n\times n}$ is the symmetric bounded positive definite inertia matrix, and n is the degree of freedom (DoF) of the robot arm; $C(q,\dot{q})\dot{q} \in \mathbb{R}^n$ denotes the Coriolis and Centrifugal force; $G(q) \in \mathbb{R}^n$ is the gravitational force; $\tau \in \mathbb{R}^n$ is the vector of control input torque; and $\tau_{ext} \in \mathbb{R}^n$ is the external force.

B. Robot Kinematic Model

1) Forward Kinematics: Each arm of iCub robot is comprised of 16 joints; however we only study the first 7 joints of shoulder, elbow and wrist, respectively. In this paper, Denavit-Hartenberg (DH) parameters are used to describe the robot kinematics model. DH parameters include joint angle θ, link offset d, link length a and link twist α[10][11]. The D-H parameters of iCub robot are listed in Table II[12].

In this paper, T_i^{i-1} is used to denote the homogeneous transformation matrix which represents the transformation from link coordinate frame {i} to frame {i-1}, that is:

$$P^{i-1} = T_i^{i-1} P^i \tag{2}$$

where P^{i-1} and P^i denote the augmented position of a point

in coordinate frame $\{i\}$ and frame $\{i-1\}$, respectively,

$$T_i^{i-1} = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & \alpha_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1}d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & -s\alpha_{i-1}d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

Here $s\theta$ and $c\theta$ present $\sin\theta$ and $\cos\theta$, respectively.

By introducing

$$T_7^0 = T_1^0 T_2^1 T_3^2 T_4^3 T_5^4 T_6^5 T_7^6, \tag{4}$$

when the 7 joint angles $(\theta_1 \cdots \theta_7)$ are known, we can calculate the Cartesian position and orientation. Because Falcon joystick only sends commands of Cartesian position, we do not consider object orientation in this paper. We define a function *fkine* to describe the relationship mentioned above:

$$X = fkine(q) \tag{5}$$

where $X \in \mathbb{R}^3$, denoting the linear position in Cartesian space, and $q \in \mathbb{R}^7$, representing angular position in joint space.

In our experiment, the human operator sends position commands to the virtual iCub robot through the Falcon joystick to control its motion. The Falcon joystick is a 3-DoF haptic device of high cost performance as shown in Fig.4. This device of parallel linkages can be used in research related to control and estimation problems of robots[14]. In [7], Falcon joystick is introduced into the virtual surgery system as the key device which can realize force feedback in the virtual environment.

Fig. 4: Force feedback device-Falcon joystick [15]

2) Inverse Kinematics: Seven joints constitute the iCub robot manipulator and every joint makes contributions to the Cartesian position of the end-effector, therefore, there is a huge computational load to solve the inverse kinematic problem to transform desired trajectory from Cartesian space to joint space. Any improper solution of inverse kinematics may cause problems of position control. In order to solve this problem, we employ the method of velocity Jacobian[16]. Its mathematical description is as follows:

$$\begin{aligned} \dot{q}_d &= J_v(q)^\dagger \dot{X}_d \\ J_v(q) &= I_{3\times 6} J(q) \end{aligned} \tag{6}$$

where

$$I_{3\times 6} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \tag{7}$$

$q \in \mathbb{R}^7$ denotes the position of seven joints in joint space, q_d denotes the desired position in joint space given the desired position in the Cartesian space, $J_v(q)^\dagger \in \mathbb{R}^{7\times 3}$ is the pseudo inverse of Jacobian matrix $J_v(q)$, $X_d \in \mathbb{R}^3$ denotes the end-effector position in Jacobian space, and $J \in \mathbb{R}^{6\times 7}$ represents the Jacobian matrix of iCub manipulator. Here, we denote the initial states of the robot arm as $q(0)$ (angles of joints) and it satisfies

$$X_d(0) = fkine(q(0)) \tag{8}$$

where $X_d(0)$ is the initial value of the desired position in Cartesian space.

Then we can obtain the solution of inverse kinematics through numerical integration (such as Runge-Kutta algorithm) of Eq. (6) under initial condition $q(0)$ as follows:

$$\hat{q}(t_f) = \int_{t_0}^{t_f} (J_v(q(t))^\dagger \dot{X}_d) dt + q(0) \tag{9}$$

for every time instant t_f.

In order to examine the precision of this method, we test it with the Falcon joystick sending command of robot hand desired position X_d in Cartesian space (with proper scaling and translation), from which we calculated $\dot{X}_d$ and use it to solve inverse kinematics using Eq. (9). In this experiment, the iCub's manipulator is supposed to move in Y direction only back and forth following the commanded position from Falcon joystick. We compare the position trajectory commanded by the Falcon joystick and the robot hand position calculated by inverse kinematics Eq. (9) and then forward kinematics Eq. (8).

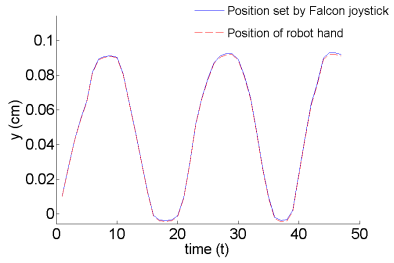


Fig. 5: Solution of Inverse Kinematics

The result is shown in Fig.5. Apparently, the error between the desired position set by Falcon joystick and the position of iCub hand calculated by kinematics is sufficiently small, i.e. the maximum of error is 1.02×10^{-3}. We have performed this experiment for 50 times and collected the experimental datas. Analysis shows that the average error is not larger than 1.1×10^{-3}. The performance of the inverse kinematics algorithm mentioned above is satisfactory.

C. Object Dynamic Model

As shown in Fig. 2, the object is modeled as a combination of spring-mass-dampers. The object will produce

slight deformation when it is handled by iCub robot's two manipulators[9]. This process can be expressed as follows:

$$\Delta_X = 2r - (X^f - X^l) \tag{10}$$

where X^f represents the position of the following manipulator's end-effector and X^l represents the leading one's.

If $\Delta_X \leq 0$, the distance between iCub robot's two manipulators is larger than the diameter of the object and the object will fall down; if $\Delta_X > 0$, from *Newton's Second Law of Motion*, we can obtain then motion equation of the mass shown in Fig.2 as follows:

$$m\ddot{X}^f = F^f + F_{dk} \tag{11}$$

where F_{dk} is the force caused by deformation of the spring and the effect of dampers, and it can be expressed as follows:

$$F_{dk} = d\dot{\Delta_X} + k\Delta_X \tag{12}$$

Here, F^f represents the force which should be applied to the mass by the following manipulator. From robot kinematics, we know that:

$$\tau^f_{ext} = (J^f)^T F^f \tag{13}$$

where $(J^f)^T$ is the transpose of Jacobian matrix of iCub following manipulator, and τ^f_{ext} is the equivalent joint torque applied on the following robot arm.

From robot dynamics, we know:

$$\tau^f = M^f(q^f)\ddot{q}^f + C^f(q^f, \dot{q}^f)\dot{q}^f + \tau^f_{ext} + G^f(q^f) \tag{14}$$

We simulate the whole iCub robot-object system described above in MATLAB/Simulink with the RVC toolbox, which is mainly developed by Peter Corke[13].

III. Force Control

A. Non-fuzzy feed-forward Compensation Controller

The PD control with feed-forward compensation consists of a linear PD feedback plus a feed-forward computation of the nominal robot dynamics along the desired joint position trajectory[17]. In this paper, we design a force feed-forward compensation controller. In this section, the controller we design is only applied to the following manipulator. From the robot dynamics, we design the control law as follows:

$$\tau^f = M^f(q^f_d)\ddot{q^f_d} + C^f(q^f_d, \dot{q}^f_d)\dot{q^f_d} + G^f(q^f_d) + \tau^f_{ext} + \tau^f_{pd} \tag{15}$$

where $\tau^f_{pd} = K_p e + K_d \dot{e}$, $e = q^f_d - q^f$, $\dot{e} = \dot{q}^f_d - \dot{q}^f$ and q^f_d is the desired joint angles of following manipulator.

In order to keep the distance between the leading manipulator and the following one, their velocity in Cartesian space should stay the same. We are able to obtain the task space velocity of the leading manipulator, $\dot{X}^l_d$, which is also the desired velocity of the following manipulator in Cartesian space, i.e., $\dot{X}^l_d = \dot{X}^f_d$. From Eq. (6), Eq. (8) and Eq. (9), using the initial value of the joint angles, $q^f(0)$, we can obtain the desired angles of the following manipulator, namely q^f_d, through numerical integration of

$$\dot{q^f_d} = J^f_v(q^f)^\dagger \dot{X}^f_d \tag{16}$$

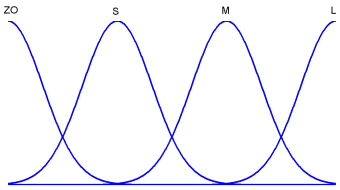


Fig. 6: Gaussian Curve Built in Membership Function

under initial condition $q^f(0)$ as follows

$$q^f_d(t_f) = \int_{t_0}^{t_f} (J^f_v(q^f)^\dagger \dot{X}^f_d)dt + q^f(0) \tag{17}$$

for every time instant t_f.The initial distance between two robot hands are assumed to be of the desired value p, i.e., $\|X^l(0) - X^f(0)\| = p$, with $p < 2r$. Once $q^f_d(t_f)$ is calculated for every time instant t_f, one may compute the control signal $\tau^f(t_f)$ via Eq. (15).

B. Fuzzy Feed-forward Compensation Controller

In the Feed-forward compensation controller, we need to select suitable control parameters K_p and K_d during the control process. However, when the external and internal environments change, the parameters may not be the optimal choice again.

Fuzzy control is a kind of computer control technology based on human language rule and fuzzy logic inference. It does not depend on the mathematical model of the system but on "fuzzy rules" which are transformed from operational experience and some known knowledge. To some extent, fuzzy control is comparable of human intelligence, belonging to intelligent control. Zedeh initiated the fuzzy set theory in 1965. After that, fuzzy logics has been applied to control area, which is the base of fuzzy control theory. In 1974, Mamdani invented the first steam engine which was based on the fuzzy control theory [18] [19]. Fuzzy control has since been quickly developed and widely used.

1) Fuzzification and Defuzzification: We select error $|e|$ in joint space and its derivative $|\dot{e}|$ as the input of the fuzzy inference. The two crisp variables must be transformed to fuzzy variables in order to match the fuzzy rule described by human expert language, that is fuzzy subset. This process is called fuzzification.

We choose Gaussian curve as the membership function of the fuzzy subset, its mathematical description is given by:

$$f(x, \sigma, c) = e^{\frac{-(x-c)^2}{2\sigma^2}} \tag{18}$$

We can select the suitable parameters, σ and c, in the Eq.(18) to express the four fuzzy subsets in the fuzzy PD controller-A(ZO), A(S), A(M) and A(L), which represent the fuzzy variable is zero, small,medium and large, respectively, as shown in Fig. 6. We can express the fuzzy set by Zedeh Method[18]:

$$A = \frac{A(ZO)}{ZO} + \frac{A(S)}{S} + \frac{A(M)}{M} + \frac{A(L)}{L} \tag{19}$$

TABLE III: Fuzzy rules for $\Delta {K_p}^{(i)}$

$\dot{e}^{(i)}$ \ $e^{(i)}$	L	M	S	ZO
L	M	S	M	M
M	L	M	L	L
S	L	M	L	L
ZO	L	M	L	ZO

TABLE IV: Fuzzy rules for $\Delta {K_d}^{(i)}$

$\dot{e}^{(i)}$ \ $e^{(i)}$	L	M	S	ZO
L	S	M	ZO	ZO
M	M	M	S	ZO
S	L	L	S	S
ZO	L	L	S	ZO

We define A as the input fuzzy set and, similarly, set B as the output fuzzy set. $f_A(u)$ and $f_B(y)$ are the membership functions of input and output. When one crisp input variable, such as u, needs to be fuzzificated, we can get its membership values (u_1, u_2, u_3, u_4) in four input fuzzy subsets-A(ZO), A(S), A(M) and A(L) through Eq.(18). According to Eq. (19), we see the variable, u, can be expressed in the form of fuzzy set as follows:

$$u = [u_1 \quad u_2 \quad u_3 \quad u_4] \tag{20}$$

After fuzzy logic inference, the output is a fuzzy set. We need to transform it to some representative value. There are a number of methods to achieve defuzzification, such as centroid, bisector and maximum. Here, we adopt centroid.

Suppose the output result is a fuzzy set, $Y = [y_1 \quad y_2 \quad y_3 \quad y_4]$, then, y_{cen} is the actual output can be expressed as follows:

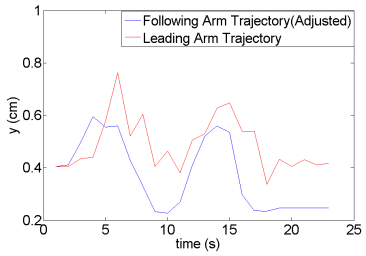
$$y_{cen} = \frac{\sum_{i=1}^{i=4} f_B(y_i) y_i}{\sum_{i=1}^{i=4} f_B(y_i)} \tag{21}$$

2) Fuzzy Rule: To design control law as given in Eq. (15), we only focus on the design of τ_{pd}^f described as below:

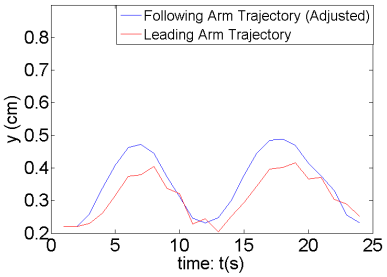
$$\tau_{pd}^f = K_p e + K_d \dot{e} \tag{22}$$

where K_d and K_p are chosen to be positive definite diagonal matrices, which means that for simplicity each component $e^{(i)}$ ($i \in \{1, 2, \cdots, 7\}$) of e requires a pair of PD controller gains $(K_p^{(i)}, K_d^{(i)})$. Obviously, Eq. (22) can be seen as a PD control term. Combining the traditional PD control algorithm and fuzzy control theory, in this paper we design a fuzzy PD controller which can adjust the parameters of the PD control term online. To this end, the fuzzy rules shown in Table III and Table IV are designed for the purpose of controlling each dimension $e^{(i)}$ of the error e according to the experience of tuning PD controllers in practice according to the error, $e^{(i)}(t)$, and the changing rate of the error, $\dot{e}^{(i)}(t)$.

In the tables above, L, M, S and ZO represent that the fuzzy variable is large, medium, small or zero, respectively.



(a) Non-fuzzy Controller



(b) Fuzzy Adaptive Controller

Fig. 7: Tracing Trajectories

Symbols K_{p0} and K_{d0} are the initial values of the PD controller parameters, ΔK_p and ΔK_d denote the modification of K_{p_0} and K_{d_0}, so the real-time parameters of PD controller, K_p and K_d are as follows:

$$\begin{aligned} K_p &= K_{p_0} + \Delta K_p \\ K_d &= K_{d_0} + \Delta K_d \end{aligned} \tag{23}$$

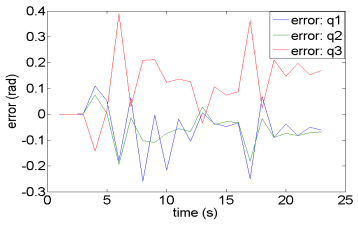
In the iCub manipulator's 7 joint angles, all of them can affect the position of the end-effector . In this experiment, we only desire to control the end-effector moving along a single axis, i.e., in the Y direction of reference coordinate system.

C. Simulation and Results

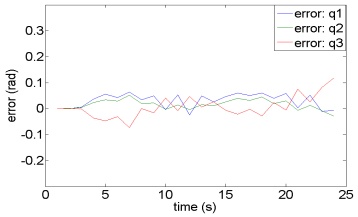
The experiment is simulated on the computer. The motion model is simplified and only one dimension of the motion is considered which means that the object is only moving in the Y direction. Noise and disturbance are also neglected. In the process of simulation, the leading arm's's Cartesian position is manipulated directly by human operator through Falcon joystick with proper scaling and translation..

The main results of the simulation are shown in Fig.7-Fig.8, where for convenience of comparison, the trajectory (in the Y axis) of the following arm is adjusted by shifting $-p$, i.e. we plot $y^f - p$ instead of y^f for ease of comparison with y^l.

From the simulation results, we can see that both fuzzy-based controller and non-fuzzy controller could achieve the goal, but with different performance.



(a) Non-fuzzy Controller



(b) Fuzzy Adaptive Controller

Fig. 8: Error Comparison in Joint Space

In Fig.7a and Fig.7b the Cartesian trajectories under different controllers are shown, respectively. The blue one denotes the path, adjusted with a shift $-p$, that the following manipulator moves along. Statistics shows that the mean of error under none-fuzzy controller is 0.117 and the variance is 0.0125; the mean of error under fuzzy controller is 0.0409 and the variance is 0.0017. The Fig.8a and Fig.8b show the tracking errors of the following arm in joint space with only 3 joints selected to be shown. Obviously, the fuzzy-based feed-forward compensation controller performs better than the non-fuzzy controller. The tracking performance under the fuzzy controller has higher precision, less overshoot and its fluctuation is less than non-fuzzy controller.

The feed-forward compensation controller's structure is relatively simple and this algorithm can be calculated faster than other controllers. Because the fuzzy-based feed-forward compensation controller can adaptively regulate parameters, it has a great advantage over the non-fuzzy controller in the transient characteristics. Overall, the fuzzy-based feed-forward compensation controller produces a better result.

IV. Discussion and Conclusion

In the experiment of this paper, one robot arm manipulated by the human operator plays the role of leading arm, and the other arm tries to adapt to the leader in the absence of the leader's exact mathematical model by using the feedback information through the object connected with the two arms.

In comparison to previous research, this paper aims to bring human intelligence into the robotic system which means that human operator will teach the iCub robot how to accomplish the task. From the preliminary experiments conducted, the merits of the introduced human-robot collaborative control model can be summarized as follows:

- The experiment conducted has illustrated that the following manipulator can successfully collaborate with the leading manipulator without complete knowledge of the leading manipulator.
- In the experiment of this paper, the system includes human operator as part of the whole closed loop, which brings human intelligence into the system and makes the whole system more flexible and reliable than the completely autonomous robot.

Acknowledgments

The authors thank Alex Smith for his help to build the iCub robot model in MATLAB with RVC Toolbox and to improve the English of the paper.

References

[1] M. L. Chen and D. J. Tao. Research and development of manipulator. *Overseas Travel and Employment*, 15(2):84, 2012.
[2] The robotcub consortium. http://www.icub.org.
[3] V. Mohan, P. Morasso, and J. Zenzeri. Teaching a humanoid robot to draw shapes. *Autonomous robots*, 31(1):21–53, 2011.
[4] M. Fumagalli, M. Randazzo, and F. Nori. Exploiting proximal F/T measurements for the iCub active compliance. *RSJ International Conference on Intelligent Robotics and Systems*, 2010.
[5] L. Chen and Z. Y. Liu. The adaptive control of coordinated motion of spacecraft attitude and its manipulator. *Control Theory and Applications*, 19(2):274–278, 2002.
[6] J. Ye, J. F. Qiao, and M. Li. A behavior-based intelligent controller for a 2-DoF manipulator. *Control Theory and Applications*, 24(3):444–448, 2007.
[7] C. Distance, A. Anglani, and F. Taurisano. Target reaching by using visual information and Q-learning controllers. *Autonomous Robots*, 9(1):41–50, 2000.
[8] S. S. Jia and G. D. Liu. Application of model based disturbance attenuator to the control of robot manipulator. *Control Engineering of China*, 14(5):544–547, 2007.
[9] A. Smith, C. Yang, H. Ma, P. Culverhouse, A. Cangelosi, and E. Burdet. Bimanual robotic manipulation with biomimetic joint/task space hybrid adaptation of force and impedance. *11th IEEE International Conference on Control & Automation*, 2014.
[10] F. F. Zhou, X. P. Fan, and Z. Ye. 3-D virtual robotic model generated by D-H parameters. *Journal of System Simulation*, 18(4):947–950, 2006.
[11] H. Y. Fang and X. Y. Liu. Parameter determination in robot kinematic model. *Journal of Machine Design*, 28(2):46–49, 2001.
[12] The forward kinematics for iCub robot. http://wiki.icub.org/wiki/ICubForwardKinematics.
[13] P. I. Corke. Arobotics toolbox for matlab. *Robotics & Automation Magazine, IEEE*, 3(1):24–32, 1996.
[14] S. Martin and N. Hillier. Characterisation of the Novint Falcon haptic device for application as a robot manipulator. *2009 Australasian Conference on Robotics and Automation*, pages 145–153, 2009.
[15] Novint technologies. http://www.novint.com/.
[16] H. B. Xin and Y. Q. Yu. Haptic interaction between human and virtual iCub robot using Novint Falcon with CHAI3D and MATLAB. *Journal of Machine Design*, 24(8):20–25, 2007.
[17] V. Santibanez and R. Kelly. PD control with feedforward compensation for robot manipulators: analysis and experimentation. *Robotica*, pages 11–19, 2001.
[18] X. M. Shi and Z. Q. Hao. Fuzzy control and MATLAB simulation. *Tsinghua University Press, Beijing Jiaotong University Press*, pages 7–9, 2008.
[19] B. G .Hu and H. Ying. Review of fuzzy PID control techniques and some important issues. *ACTA Automatica Sinica*, 27(4):567–584, 2001.