Intention Detection in Upper Limb Kinematics Rehabilitation Using a GP-based Control Strategy

Yongzhuo Gao[1], Yanyu Su[1], Wei Dong[1], Zhijiang Du[1], Yan Wu[2]

***Abstract*— In robot-assisted upper limb rehabilitation, detecting the intentions of hemiplegic patients is essential towards assisting the patients to actively exercise instead of driving passive motions. Many interactive channels, such as voice, EMG and EEG, have been studied to estimate the motion intentions. However, limitations of these techniques, such as high complexity, have constrained their applications in practice. In this paper, we integrate a virtual environment and a low-cost motion sensor into a novel control strategy to detect motion intentions for a rehabilitation robot. Several bimanual motion sequences are intuitively programmed by a professional therapist for subjects to repeat. The strategy uses the unaffected arm and the programmed motion sequence to estimate the motion intentions of the affected arm. We adopt this strategy in Mirror Therapy, a widely-practised therapeutic intervention method. Experiments have been conducted to validate the control strategy.**

I. Introduction

Many stroke survivors suffer from hemiplegia, which causes the loss of one side limb motor functions. Many upper limb rehabilitation robotic systems have been studied to assist these patients in rehabilitation training, such as the MIT-MANUS[1], the MIME[2] and the REHAROB[3]. Theoretically, the patient can decide on the movements he/she wants to do and the robot would be able to detect those intentions in order to assist his/her affected limb as needed. Different interactive channels, such as, voice command[4], surface electromyography(sEMG)[5] and electroencephalography(EEG)[6], have been studied to achieve this target. However, they are hindered by many limitations, such as low communication bandwidth, high complexity and high costs, in practice. Recently, several new ideas for intention detection for upper limb rehabilitation have been proposed. New sensors, such as eye-tracker, the Microsoft Kinect, combined with virtual environment are used to tackle this problem. However, sensor fusion is needed for the algorithm[7] or rehabilitation robot is not utilised in these technologies[8], [9]. Thus, we propose a control strategy to detect the motion intentions applied to the rehabilitation robot together with a Virtual Reality technology and a low-cost human motion sensor, the Kinect.

We focus on a type of training program, shown as Figure 1, that asks the patient to watch a virtual human and mimic a motion sequence demonstrated by the therapist. We assume

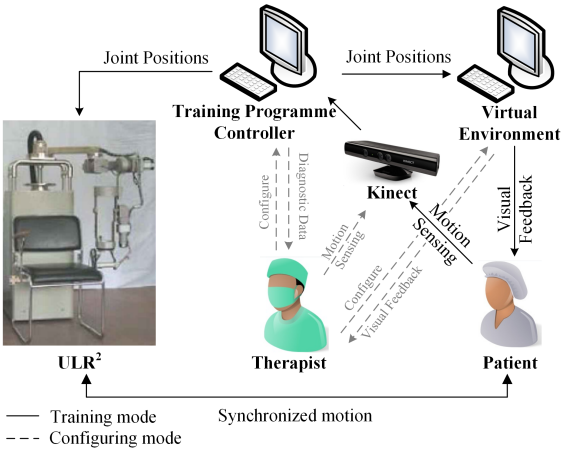

Fig. 1: The concepts of the strategy

that the patient will *intentionally* try to *synchronously* move both his/her unaffected arm and affected arm to follow the programmed motion sequence. Therefore, we can use the motion of the unaffected arm to estimate the time step of the programmed motion sequence, at which the patient is moving, then use the programmed motion of the affected arm at the same time step as the training input of the patient's motion intention.

Several challenges have to be addressed to implement the proposed strategy: (1) how to map the motion from the virtual therapist to the patient; (2) how to estimate which time step the patient is moving at; (3) how to map the motion from the therapist or the patient to the virtual therapist and the robot. In this work, we propose using Gaussian Process Regression to learn the Patient Model that maps the motion of the virtual therapist to the patient, and estimate the time step by solving a optimal control problem to find the motion in the programmed motion sequence that is most likely to be the current motion of the patient using the Patient Model. As the problem of generating robot motions from captured human motions has been well studied in the literature [10][11], we will adopt a naive kinematics based method to solve this problem.

In order to investigate the scenario of applying the proposed method in the clinic setup, we adopted the method of Mirror Therapy where the patient moves both arms as symmetrically as possible while he/she watches the mirror image as if the affected arm behind the mirror is unaffected[12]. The proposed controller is an adequate tool for implement the Mirror Therapy for several reasons: 1) As the accuracy

[1]Yongzhuo Gao, Yanyu Su, Wei Dong, Zhijiang Du are with the State Key Laboratory of Robotics and System, Harbin Institute of Technology, China. {gaoyongzhuo; suyanyucn; dongwei; duzj01}@hit.edu.cn

[2]Yan Wu is with the A*STAR Institute for Infocomm Research, Singapore. wuy@i2r.a-star.edu.sg

requirements is not restrict, many studies have successfully applied the Kinect for building virtual environment for the Mirror Therapy[13][14][9]. 2) As none of these studies uses robot to help the patient in moving the affected arm, our method will extend the usability.

The rest of the paper is organized as follows: We will introduce the platform in Section II, explain the method of the control strategy in Section III and present an experiment to validate the proposed strategy in Section IV.

II. Platform

We have set up a platform to implement our control strategy, including *the Upper Limb Rehabilitation robot*, *the virtual environment* and *the Kinect*. The upper limb rehabilitation robot(HIT-ULR^2) consists of a robotic exoskeleton, a mounting rack and a control system [15][16], as shown in Figure 1. The robot has five Degrees-of-Freedom (DoF). The range of motion(RoM) for each exoskeletal joint is limited by mechanical end stop for safety. Torque sensors are equipped at each joint to acquire interactive forces between the patient and the robot.

The virtual environment is used for visual feedback. In our previous work [17], we implemented the virtual environment with Gazebo, which employs a graphics engine and a physics engine to carry out real-time dynamic simulation of rigid bodies.

III. Methodology

The concept of the proposed robotic system for upper limb rehabilitation is shown in Figure 1. The system consists of four functional modules, including the HIT-ULR^2, the Kinect, the virtual environment and the controller. Its control strategy has two modes: the training mode and the configurature mode.

In the configuration mode, the therapist can configure the controller to customise the training programs through the Graphic User Interfaces (GUIs) and the Kinect: For example, the therapist may modify the virtual environment through the GUI of Gazebo (the simulator adopted for the virtual environment) and program the training motions using program by demonstration (PbD) techniques with the Kinect.

In the training mode, the patient wears the rehabilitation robot on the affected arm and observes the virtual environment to carry out the training programs. The motion of the patient is sensed by the Kinect. Following certain training program, the controller takes the motions of the unaffected arm to generate the motion of the affected arm before updating the virtual environment and sending the motion commands to the HIT-ULR^2. In this paper, we have implemented two training programs: One is the RoM Exercise, where the controller generates the motion with reference to the programmed motion sequence; The other one is the Mirroring Exercise, where the controller generates the motion directly with reference to the unaffected arm.

The control strategy consists of several modules, as shown in Figure 2, where three modules shall be emphasized at the beginning: 1) the virtual therapist who is capable to carry out accurate movement to illustrate the programmed motion sequence, 2) the patient model which maps the motion of the virtual therapist to the patient, 3) the transformation which maps the motion of a human to the virtual therapist and the robot. In the following of this section, we will detail each module. At the same time, we will discuss how to adopt the proposed method for Mirror Therapy.

A. Kinematics

1) Kinematic Model of Kinect Skeleton: The acquired data from the Kinect is the positions of the key points of the skeleton

$$\boldsymbol{P}_k = [\boldsymbol{p}_{k_1}\ \boldsymbol{p}_{k_2}\ \dots\ \boldsymbol{p}_{k_{10}}] \in \mathbb{R}^{10\times 3},$$

with $\boldsymbol{p}_{k_i} \in \mathbb{R}^3, i = 1, 2 \dots 10$ denoting the position of each key point k_i with respect to the kinect sensor frame $\{\boldsymbol{N}\}$.

To map this information to robot motions, the kinematic model of the skeleton is built as shown in Figure 3(a), where nine frames (from $\{\boldsymbol{K}_1\}$ to $\{\boldsymbol{K}_9\}$) are defined. The origin of each frame is defined at the respective key points. The x-axis of the reference frame $\{\boldsymbol{K}_1\}$ is defined by

$$\boldsymbol{x}_{k_1} = \boldsymbol{n}(\boldsymbol{p}_{k_2} - \boldsymbol{p}_{k_6}),$$

with the function $\boldsymbol{n}(\bullet)$ generating the normalized vector, while the y-axis is defined as upward positive and perpendicular to x-axis. The orientation of the shoulder frames, $\{\boldsymbol{K}_2\}$ and $\{\boldsymbol{K}_6\}$, are identical to that of the frame $\{\boldsymbol{K}_1\}$. The y axis of other frames are defined by

$$\boldsymbol{y}_{k_i} = \boldsymbol{n}(\boldsymbol{p}_{k_{i-1}} - \boldsymbol{p}_{k_i}),$$

while the z-axis of other frames are defined by

$$\boldsymbol{z}_{k_i} = \boldsymbol{n}(\boldsymbol{x}_{k_0} \times \boldsymbol{y}_{k_i}),$$

with $i = 2, 3, 4, 6, 7, 8$ as the index of the frames. We denote the kinematic model described above into a compact form

$$\boldsymbol{T}_{k_i} = f_{k_i}(\boldsymbol{P}),$$

with $\boldsymbol{T}_{k_i} \in \mathbb{R}^{4\times 4}$ denoting the transform matrix of frame $\{\boldsymbol{K}_i\}$.

2) Kinematic Model of HIT-ULR^2: We adopt the Denavit-Hartenberg(D-H) notation to describe the kinematics model of the HIT-ULR^2. We denote the forward kinematics model as

$$\boldsymbol{T}_{r_i} = f_{r_i}(\boldsymbol{q}, \boldsymbol{\rho}),$$

where $i = 1, 2, \dots, 5$ is the index of the frames, $\boldsymbol{T}_{r_i} \in \mathbb{R}^{4\times 4}$ denotes the transform matrix of frame $\{\boldsymbol{R}_i\}$, $\boldsymbol{q} = [q_1, q_2, \dots, q_5] \in \mathbb{R}^5$ denotes the positions of the joints, $\boldsymbol{\rho} = [l_1, l_2, l_3] \in \mathbb{R}^3$ denotes the length parameters of the D-H notation. The parameters of the D-H notation are shown in Table I with the left arm respectively shown in Figure 3(c).

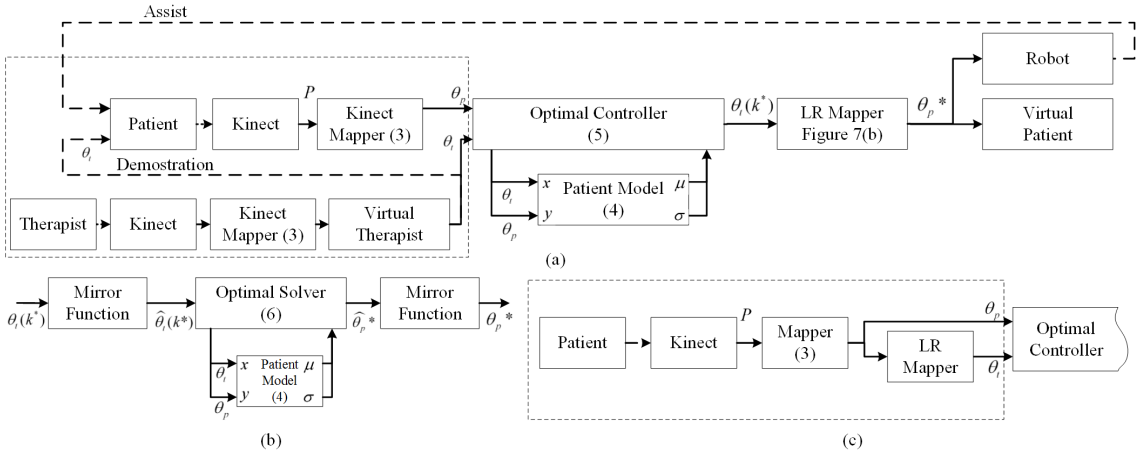

Fig. 2: The block diagram of the control strategy is shown in (a). The detailed structure of the module LR Mapper is shown in (b). The modification made to the controller for the mirroring training program is shown in (c). Important equations are marked on the respective blocks, for example, referring to (5) for the optimal controller.

TABLE I. The D-H Notations of HIT-ULR^2

Link	$\alpha(^o)$	$a(mm)$	$\theta_0(^o)$	$d(mm)$	$limit(^o)^{a,b}$
1	90	l_1	0	0	$0 \sim 90$
2	0	0	0	0	$-45 \sim 180$
3	-90	0	-90	0	$0 \sim 135$
4	90	0	0	l_2	$-90 \sim 90$
5	0	l_3	90	0	$-45 \sim 45$

[a] values for the left arm, using opposite values for the right arm;
[b] each value has been added by the respective value of θ_0.

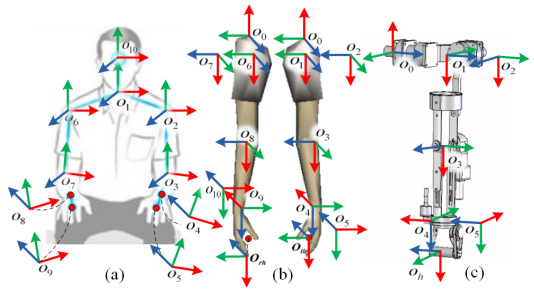

Fig. 3: The kinematic model of the Kinect skeleton, the virtual human and the HIT-ULR^2 are respectively shown in (a), (b) and (c).

3) Kinematics Model of the Virtual Human: We build the kinematics model of the virtual human to be identical to that of the HIT-ULR^2 as shown in Fig3(b). We denote the forward kinematics as

$$\boldsymbol{T}_{v_i} = f_{v_i}(\boldsymbol{\theta}, \boldsymbol{\rho}),$$

where $i = 1, 2, \ldots, 10$ is the index of the frames, $\boldsymbol{T}_{v_i} \in \mathbb{R}^{4\times4}$ denotes the transform matrix of frame $\{\boldsymbol{V}_i\}$, $\boldsymbol{\theta} = [\theta_1, \theta_2, \ldots, \theta_{10}] \in \mathbb{R}^{10}$ denotes the joint position of the virtual human.

B. Mapping Kinect Skeleton to Robot and Virtual Human

Since the mechanical configurations of HIT-ULR^2 and the Kinect skeleton are different, we map the Kinect skeleton to the HIT-ULR^2 by minimizing the error of the directional vectors between the Kinect skeleton and the HIT-ULR^2 against the joint limits of the HIT-ULR^2. Therefore, the optimization problem for the left arm can be defined by

$$\boldsymbol{q}_k = \underset{\boldsymbol{q}}{\operatorname{argmin}} \; ||T_t * \boldsymbol{x}_{r_3}(\boldsymbol{q}, \boldsymbol{\rho}) - (-\boldsymbol{y}_{k_3})||, \tag{1}$$

such that

$$||\boldsymbol{T}_t * \boldsymbol{x}_{r_2}(\boldsymbol{q}, \boldsymbol{\rho}) - (-\boldsymbol{y}_{k_2})|| = 0$$
$$||\boldsymbol{T}_t * \boldsymbol{x}_{r_{lh}}(\boldsymbol{q}, \boldsymbol{\rho}) - (-\boldsymbol{y}_{k_5})|| = 0$$
$$(\boldsymbol{q})_{min} \leq \boldsymbol{q} \leq (\boldsymbol{q})_{max}$$

and for the right arm

$$\boldsymbol{q}_k = \underset{\boldsymbol{q}}{\operatorname{argmin}} \; ||T_t * \boldsymbol{x}_{r_8}(\boldsymbol{q}, \boldsymbol{\rho}) - (-\boldsymbol{y}_{k_7})||, \tag{2}$$

such that

$$||\boldsymbol{T}_t * \boldsymbol{x}_{r_6}(\boldsymbol{q}, \boldsymbol{\rho}) - (-\boldsymbol{y}_{k_6})|| = 0$$
$$||\boldsymbol{T}_t * \boldsymbol{x}_{r_{rh}}(\boldsymbol{q}, \boldsymbol{\rho}) - (-\boldsymbol{y}_{k_9})|| = 0$$
$$(\boldsymbol{q})_{min} \leq \boldsymbol{q} \leq (\boldsymbol{q})_{max}$$

where $(\boldsymbol{q})_{min}$ and $(\boldsymbol{q})_{max}$ respectively denotes the lower and the upper positional limitations of the robot joints, $\boldsymbol{T}_t \in \mathbf{R}^{4\times4}$ is the transform matrix that maps the frame $\{\boldsymbol{O_0}\}$ of the HIT-ULR^2 to the respective shoulder frame of the Kinect skeleton (the frame $\{\boldsymbol{K_2}\}$ or the frame $\{\boldsymbol{K_6}\}$), $\boldsymbol{x}_{r_2}$, $\boldsymbol{x}_{r_3}$ and $\boldsymbol{x}_{r_h}$ respectively denote the x-axis of the frames of the HIT-ULR^2,

$$\boldsymbol{x}_{r_i} = \boldsymbol{v}_x(f_{r_i}(\boldsymbol{q}, \boldsymbol{\rho})),$$

where $\boldsymbol{v}_x(\bullet)$ denotes the function fetching the x directional vector from a transform matrix.

As the problem defined in (1) and (2) are continuous-space nonlinear optimal problems, we adopt the interior-point method, a certain class of algorithms that solves linear and nonlinear convex optimization problems, to solve it and denote it as

$$q_i = g_{r_i}(\boldsymbol{P}, \boldsymbol{\rho}),$$

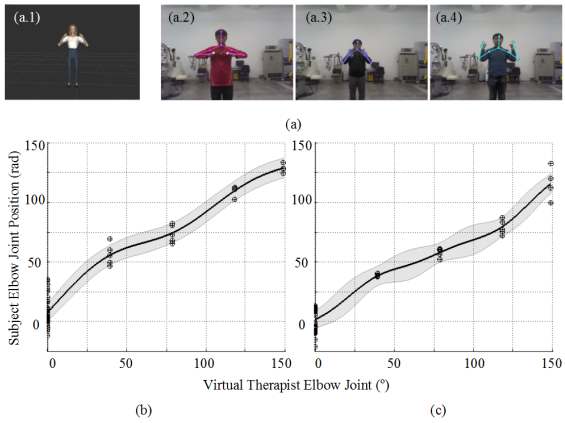

Fig. 4: Given the same demonstration for calibrating the patient model of the elbow flexion/extension joint, different subject may perform different movement as shown in (a). The resulted user model of the subject (a.1) and (a.2) are respectively shown in (b) and (c), where the solid line is the predictive mean of patient model and the shaded area represents the predictive variance.

where $i = 1, 2, \ldots, 5$ is the index of the frames of the robot.

As the kinematics model of the virtual human is identical to the HIT-ULR^2, the mapping method for the HIT-ULR^2 is applicable to map the Kinect skeleton to the virtual human by mapping the Kinect skeleton to the left arm and the right arm of the virtual human using (1) and (2). Therefore, we can use the joint positions of the respective virtual human to describe the Kinect skeleton

$$\theta_i = g_{k_i}(\boldsymbol{P}, \boldsymbol{\rho}), \tag{3}$$

where $i = 1, 2, \ldots, 10$ is the index of the virtual human joints.

C. Virtual Therapist and its Programming

We use a virtual human, named as the virtual therapist, to demonstrate the programmed motion sequence, so the patient could mimic it. The therapist could modify the virtual therapist by specifying its length parameters $\boldsymbol{\rho}_t$. The motion of the virtual therapist is described by the joint positions

$$\boldsymbol{\theta}_t = [\theta_{t_1}, \theta_{t_2}, \ldots, \theta_{t_{10}}].$$

We let the therapist program the motion sequence by recording it through the Kinect and use the method described in (3) to generate the motions of the virtual therapist from the acquired sequence of the Kinect skeleton.

D. Virtual Patient

We use a virtual human, named as virtual patient, to represent the patient in the virtual environment. The length parameters of the virtual patient $\boldsymbol{\rho}_p$ are estimated with the distances between the respective key points of the Kinect Skeleton. In this work, we describe the motions of the patient through the motions of the virtual patient generated from the Kinect Skeleton with (3), and denote it with

$$\boldsymbol{\theta}_p = [\theta_{p_1}, \theta_{p_2}, \ldots, \theta_{p_{10}}].$$

E. Patient Modeling with Gaussian Process

As shown in Figure 4, different patients may perform different motion when the virtual therapist demonstrates the same sequence. In order to describe this phenomenon, we propose to identify the relationship between the motion of the virtual therapist and the motion of the patient by Gaussian Process regression (GPR).

For given elements $x \in \boldsymbol{X}$, a GP is specified by its mean function,

$$m(x) = \boldsymbol{E}[h(x)],$$

and its covariance function

$$k(x, x') = \boldsymbol{E}[(h(x) - m(x))(h(x') - m(x'))].$$

In contrast to other regression methods, GPR provides predictive distributions (instead of point predictions) and is able to learn the output noise from training data through maximum-likelihood maximisation. These features make GPR attractive for identifying the relationship between the given instruction and the subject reaction [18].

We adopt a GP for each joint of the virtual therapist and describe the motions of the patient through the virtual patient, therefore, the training data for our model is

$$D_i \equiv \{x_t = \theta_{t_i}, y_t = \theta_{p_i}\},$$

where $i = 1, 2, \ldots, 10$ is the index of the joints of the virtual therapist, x_t is the input, y_t is the output (or target).

After observing the experiment results, we have chosen the quadratic function as the mean function, and the squared exponential function as the covariance function. Since the GPR outputs predictive distributions, it is also capable to provide the likelihood of a test point once given a likelihood function

$$p_i(y|_{=y_s}, x|_{=x_s}) = like(GP(x_s, y_s)), \tag{4}$$

where $\{x_s, y_s\}$ denotes the test point, $like(\bullet)$ denotes the likelihood function, $GP(\bullet)$ denotes the prediction function of the Gaussian Process regression.

F. Optimal Controller

As shown in Figure 2, given the motion of the patient $\boldsymbol{\theta}_p$ and the programmed sequence of the virtual therapist $\boldsymbol{\Theta}_t = [\boldsymbol{\theta}_t(1), \boldsymbol{\theta}_t(2), \ldots, \boldsymbol{\theta}_t(N)]$ of length $N \in \mathbb{N}$, the optimal controller estimates which time step of the programmed sequence $k^* \in [1, 2, \ldots, N]$ the patient has moved at, and outputs the respective movement of the virtual therapist $\boldsymbol{\theta}_t(k^*)$.

We consider this problem as a optimal control problem. At each control step, the controller should solve the problem to find the time step of the programmed sequence k^* whose respective motion $\boldsymbol{\theta}_p(k^*)$ is most similar to the motion of the patient $\boldsymbol{\theta}_p$ according to the patient model defined in (4). Therefore, the optimal control problem is defined by

$$\boldsymbol{\theta}_t(k^*) = \underset{\boldsymbol{\theta}_t(k)}{\operatorname{argmax}} \prod_{i \in \boldsymbol{I}} p_i(y|_{=\theta_{p_i}}, x|_{=\theta_{t_i}}), \tag{5}$$

such that

$$\boldsymbol{I} = \begin{cases} \{1,2,3,4,5\} & \text{with unaffected left arm} \\ \{6,7,8,9,10\} & \text{with unaffected right arm} \end{cases}$$
$$\theta_{p_i} \in \boldsymbol{\theta}_p$$
$$\theta_{t_i} \in \boldsymbol{\theta}_t(k)$$
$$k \in \{k_c, k_c+1, \ldots, k_c+N_k\}$$

where $N_k \in \mathbb{N}$ is a parameter specifying how many steps the controller considers, $k_c \in [1,2,...,N]$ is the time step of the programmed sequence the patient has already moved at (the result of (5) at last control step) and is initialized to 1 when the virtual therapist starts to demonstrate the programmed sequence. We solve the optimal control problem by naively traversing the problem space.

G. Generating Motion of Affected Arm for Virtual Patient

The block diagram of generating the motion of the affected arm for the virtual patient is shown in Figure 2(b). As the patient model of the affected arm cannot generate the motions of a unaffected arm, we adopt three steps to generate the motions for the virtual patient using the patient model of the unaffected arm: 1) map the motion of the affected arm to the unaffected arm; 2) use the patient model of the unaffected arm to generate the motion for the virtual patient; 3) map the result back to the affected arm.

The translation of the motions from the affected arm to the unaffected arm is gained from the kinematic model shown in Figure 3 as

$$\hat{\boldsymbol{\theta}}_t(k^*) = \begin{cases} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{T}_a \end{bmatrix} \boldsymbol{\theta}_t(k^*), & \text{when left arm unaffected} \\ \begin{bmatrix} \boldsymbol{T}_a & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \boldsymbol{\theta}_t(k^*), & \text{when right arm unaffected} \end{cases}$$

where the translation matrix $\boldsymbol{T}_a$ is defined by

$$\boldsymbol{T}_a = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Given the translated motion of the virtual therapist $\hat{\boldsymbol{\theta}}_t(k^*)$, we generate the motion of the virtual patient by solving the following optimal problem with the interior-point method

$$\hat{\theta}^*_{p_i} = \underset{\theta_{p_i}}{\operatorname{argmax}}\ p_j(y|_{=\theta_{p_i}}, x|_{=\theta_{t_i}}), \tag{6}$$

such that

$$\theta_{t_i} \in \hat{\boldsymbol{\theta}}_t(k^*)$$
$$(\theta_{p_i})_{min} \le \theta_{p_i} \le (\theta_{p_i})_{max}$$
$$(i,j) = \begin{cases} \left\{ \begin{array}{l} (1,1),(2,2),(3,3),(4,4),(5,5), \\ (6,1),(7,2),(8,3),(9,4),(10,5) \end{array} \right\} \\ \qquad \text{when the left arm is unaffected} \\ \left\{ \begin{array}{l} (1,6),(2,7),(3,8),(4,9),(5,10), \\ (6,6),(7,7),(8,8),(9,9),(10,10) \end{array} \right\} \\ \qquad \text{when the right arm is unaffected} \end{cases}$$

where $(\theta_{p_i})_{min}$ and $(\theta_{p_i})_{max}$ respectively denotes the lower and the upper positional limitation of the virtual patient joint, (i,j) denotes the coupled choice of the joint and the patient model, for example $(1,6)$ and $(8,3)$ respectively means using the patient model $p_6()$ for the motion of θ_{p_1} and the patient model $p_3(\bullet)$ for the motion of θ_{p_8}.

We then translate the generated motion $\hat{\boldsymbol{\theta}}^*_p$ back before sending it to the position controller of the HIT-ULR^2 and the virtual patient

$$\boldsymbol{\theta}^*_p = \begin{cases} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{T}_a \end{bmatrix} \hat{\boldsymbol{\theta}}^*_p, & \text{when left arm is unaffected} \\ \begin{bmatrix} \boldsymbol{T}_a & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \hat{\boldsymbol{\theta}}^*_p, & \text{when right arm is unaffected} \end{cases}$$

H. Adapting the Controller to Mirroring Training Program

We simply adopted the proposed controller to mirroring motions by using its own motion as the reference motions as shown in Figure 2(c). In this mode, the affected arm will moves as the same as the unaffected arm which allows the patient to freely manipulate the objects in the virtual environment.

IV. Experiments and Results

In order to verify the system for clinical application, two experiments have been conducted, namely the Range of Motion Exercise and Mirror Therapy. The first experiment is carried out without the HIT-ULR^2 to validate the proposed strategy while the subjects are asked to wear the HIT-ULR^2 in the second experiment to investigate effect of the strategy applied in practice.

A. The Experiment Setup

The virtual environment and the motion sequence used in the experiments were created and programmed by a skilled therapist. Before the experiments, each subject was asked to take the calibration motions for ten times to generate the training data set of the patient model. To become familiar with the robot and the experiment task, each subject did the whole task once before the experiment.

B. Range of Motion(RoM) Exercise Experiment

In this experiment, the therapist created three unsymmetrical motion sequences based on RoM Exercise for the subjects to mimic, as shown in Figure 5. Ten healthy subjects (five males and five females) were asked to mimic three motion sequences without wearing the robot to validate the proposed control strategy. Without losing the generality, we assumed the left arm of the subjects were affected to configure the controller. However, as the healthy subjects would actually complete their motion intentions with the "affected" arm, we used these motions as the ground truth to verify the respective prediction of the proposed control strategy. We define the prediction error, that is, the error between the positions of the patient (in the form of the virtual human joint positions) and the predicted positions for the virtual patient

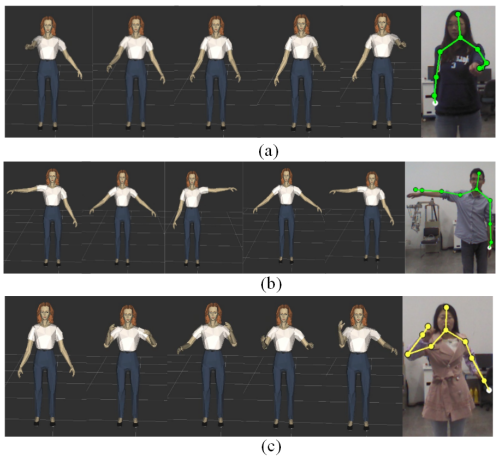

Fig. 5: Three motion sequences used in the Range of Motion Exercise experiment.

generated by the proposed control strategy to evaluate the performance

$$m_p = \sum_{i=1,2,3} \frac{1}{3} |\theta^*_{p_i} - \theta_{p_i}|$$

where θ_{p_i} is the virtual human joint positions directly mapped from the actual human motion. The wrist joints are not considered here due to the noise of the Kinect.

C. Mrirroring Training Experiment

In this experiment, every subject is asked to wear the HIT-ULR^2 and mimic three symmetrical RoM Exercise motion for each twice. In the first time the robot is set in passive mode without the proposed strategy while in active mode applied with the strategy in the second time. We design a questionnaire asking the ten subjects to score how they feel the follow of the robot when they move. The score ranges from 1 to 5, which could investigate the effect on subjects in human-robot interaction with the strategy that integrate the robot, Kinect and the virtual evironment. The results are partly shown in Table II focuses on the follow effect in several stages of the motions.

The mirroring training demo is shown in Figure 7. A female subject is carrying out a training exercise using the robotic rehabilitation system with virtual environment and motion sensing: The subject observes the virtual environment ② and moves her unaffected arm (right arm); Her motion is captured by the Kinect ③ to generate the motion of her affected arm (left arm) according to the requirements of the training exercise; The control program takes the motion of the subject and maps the virtual environment to generate motion commands of HIT-ULR^2 ①.

D. The Result

The statistical result of thirty trials in the RoM Exercise experiment is shown in Figure 6(a)(b) in box plot. Some abnormal points are also observed in the data shown as red crosses, which occurs when the patient changes his/her motion direction. The t-test on the results of all the trails

TABLE II. The Result of the Questionnaire(AVG/STDEV)

Did you feel a good follow during the following stages?						
	Shoulder-Adb/Add		Shoulder-Flx/Ext		Elbow-Flx/Ext	
STAGES	passive	active	passive	active	passive	active
Pre-raise[c]	2.3/0.9	3.9/0.5	2.1/0.8	3.9/0.8	2.0/0.6	3.9/0.7
Raise	3.4/0.7	4.6/0.5	3.0/0.6	4.6/0.5	2.2/0.7	4.9/0.3
Level	3.2/0.7	4.7/0.4	2.5/0.5	4.7/0.4	2.8/0.7	4.8/0.4
Pre-drop	1.9/0.7	3.5/0.7	1.6/0.5	4.0/0.6	1.7/0.6	4.3/0.6
Drop	3.0/0.8	4.2/0.4	2.6/0.5	4.7/0.4	2.1/0.7	4.6/0.5
Stop	2.1/0.7	4.1/0.7	1.8/0.4	4.4/0.7	1.5/0.5	4.4/0.7

[a] Ten subjects have completed the questionnaire.
[b] Full score is 5.
[c] Pre-raise refers to stage of Begin Raise, as the same to Pre-drop.

shows that the prediction error is significantly less than the error of the pre-programmed motion sequence with the p-value $p < 10^{-4}$. An example of the pre-programmed motion, the prediction of the proposed control strategy and the ground truth is shown in (c) with the respective prediction error and the error of the pre-programmed motion is shown in (d). The t-test on the result of the trail shows that the following error of each individual joint is significantly less than 5^o, as the point-to-point accuracy of the Kinect is reported around 2 cm (2.8^o with 40 cm radius) [19].

$$m_e = \sum_{i=1,2,3} \frac{1}{3} |q_i - \theta_j|$$

where, j is selected to the respective joint of the virtual patient. The wrist joints are also not considered here due to the noise of the Kinect.

While testing the strategy with HIT-ULR^2, all the subjects experience a clear auxiliary force in the active mode against the passive mode, according to the Table II. The scores in the active mode are all higher than those in the passive mode. However, the scores of the stages "Begin to Raise" and "Begin to Drop" are a bit lower. They would feel hindrance when moving too fast because of the speed limit of the robot. In fact, some patients lose the function of their forearm but not that of shoulder, so they would possibly try hard to move their upperarm than forearm. In some case, this would cause injury. In consideration of the protection, we have to limit the speed and the range of the motions, which would result in some hindrance.

V. Conclusion

In this paper, we proposed a control strategy applied to the Upper Limb Rehabilitation Robotic System and implemented it with the virtual environment technique and the motion sensing device. We addressed the problem of detecting the motion intention of the patients and focused on the training program where the patient mimics a motion sequence demonstrated by the therapist. With the assumption that patients would intentionally try to synchronously move both his/her unaffected arm and affected arm to follow the programmed motion sequence, the strategy predicted the motion intention of the affected arm based on the motion of the unaffected arm. Our experimental results have shown that the proposed control strategy could predict the motion intentions of patients with the prediction error averagely less

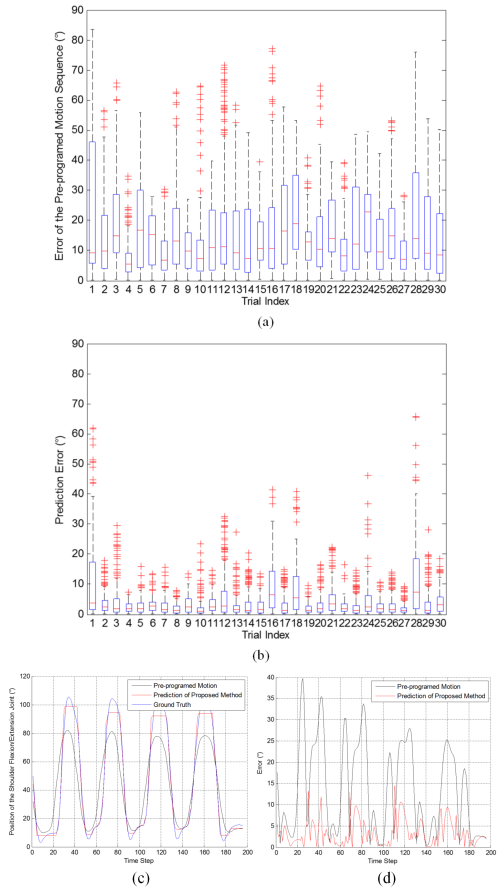

Fig. 6: The box plot of the Range of Motion Exercise experiment result is shown in (a)(b). An example of a trail is detailed in (c)(d).

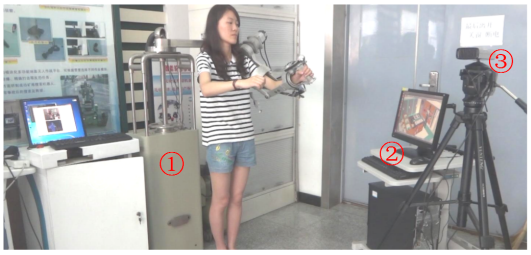

Fig. 7: The Mirroring Training Demo

than 12^o and perform a good follow and effective auxiliary force at most time.

Looking forward, we envision several improvements that would increase the utility of the control strategy. One is to integrate the force sensor and the impedance controller into the system to build a force feed-back training system. Another emergent improvement is to evaluate the system with stroke subjects. We will conduct medical experiments on the proposed to investigate how the system would work with stroke patients.

REFERENCES

[1] N. Hogan, H. I. Krebs, J. Charnnarong, P. Srikrishna, and A. Sharon, "MIT-MANUS: a workstation for manual therapy and training. I," in *Robot and Human Communication, 1992. Proceedings., IEEE International Workshop on*, pp. 161–165, IEEE, 1992.

[2] P. S. Lum, C. G. Burgar, M. der Loos, P. C. Shor, M. Majmundar, and R. Yap, "The MIME robotic system for upper-limb neuro-rehabilitation: results from a clinical trial in subacute stroke," in *Rehabilitation Robotics, 2005. ICORR 2005. 9th International Conference on*, pp. 511–514, IEEE, 2005.

[3] A. Toth, G. Fazekas, G. Arz, M. Jurak, and M. Horvath, "Passive robotic movement therapy of the spastic hemiparetic arm with REHAROB: report of the first clinical test and the follow-up system improvement," in *Rehabilitation Robotics, 2005. ICORR 2005. 9th International Conference on*, pp. 127–130, IEEE, 2005.

[4] D. E. Barkana, "Towards intelligent robot-assisted rehabilitation systems," *International Journal of Systems Science*, vol. 41, no. 7, pp. 729–745, 2010.

[5] T. Lenzi, S. M. M. De Rossi, N. Vitiello, and M. C. Carrozza, "Intention-based EMG control for powered exoskeletons," *Biomedical Engineering, IEEE Transactions on*, vol. 59, no. 8, pp. 2180–2190, 2012.

[6] K. K. Ang, C. Guan, K. Sui Geok Chua, B. T. Ang, C. Kuah, C. Wang, K. S. Phua, Z. Y. Chin, and H. Zhang, "A clinical study of motor imagery-based brain-computer interface for upper limb robotic rehabilitation," in *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, pp. 5981–5984, IEEE, 2009.

[7] D. Novak and R. Riener, "Enhancing patient freedom in rehabilitation robotics using gaze-based intention detection," *IEEE International Conference on Rehabilitation Robotics*, 2013.

[8] O. Erazo, J. A. Pino, R. Pino, A. Asenjo, and C. Fernández, "Magic mirror for neurorehabilitation of people with upper limb dysfunction using kinect," in *System Sciences (HICSS), 2014 47th Hawaii International Conference on*, pp. 2607–2615, IEEE, 2014.

[9] A. V. Dowling, O. Barzilay, Y. Lombrozo, and A. Wolf, "An Adaptive Home-Use Robotic Rehabilitation System for the Upper Body," *IEEE Journal of Translational Engineering in Health and Medicine*, vol. 2, no. April, pp. 1–10, 2014.

[10] D. Matsui, T. Minato, K. F. MacDorman, and H. Ishiguro, "Generating natural motion in an android by mapping human motion," in *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pp. 3301–3308, IEEE, 2005.

[11] M. J. Gielniak and A. L. Thomaz, "Spatiotemporal correspondence as a metric for human-like robot motion," in *Proceedings of the 6th international conference on Human-robot interaction*, pp. 77–84, ACM, 2011.

[12] E. L. Altschuler, S. B. Wisdom, L. Stone, C. Foster, D. Galasko, D. M. E. Llewellyn, and V. S. Ramachandran, "Rehabilitation of hemiparesis after stroke with a mirror," *The Lancet*, vol. 353, no. 9169, pp. 2035–2036, 1999.

[13] C. Robertson, L. Vink, H. Regenbrecht, C. Lutteroth, and B. C. Wünsche, "Mixed reality Kinect Mirror box for stroke rehabilitation.," in *IVCNZ*, pp. 231–235, 2013.

[14] L. Y. Joo, T. S. Yin, D. Xu, E. Thia, P. F. Chia, C. W. K. Kuah, and K. K. He, "A feasibility study using interactive commercial off-the-shelf computer gaming in upper limb rehabilitation in patients after stroke," *Journal of rehabilitation medicine*, vol. 42, no. 5, pp. 437–441, 2010.

[15] Q. Li, D. Wang, Z. Du, and L. Sun, "A novel rehabilitation system for upper limbs.," *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, vol. 7, pp. 6840–6843, 2005.

[16] L. Qingling, K. Minxiu, and S. Lining, "A novel 5-DOF exoskeletal rehabilitation robot system for upper limbs," *High technology letters*, vol. 15, no. 3, pp. 245–249, 2009.

[17] Z. Du, Y. Sun, Y. Su, and W. Dong, "A ROS/Gazebo Based Method in Developing Virtual Training Scene for Upper Limb Rehabilitation," in *IEEE International Conference on Progress in Informatics and Computing (PIC 2014), Shanghai, China*, pp. 307–311, 2014.

[18] Y. Su, Y. Wu, H. Soh, Z. Du, and Y. Dimiris, "Enhanced Kinematic Analysis Model for Dexterous Manipulation with Underactuated Hand," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013.

[19] K. Khoshelham, "Accuracy analysis of kinect depth data," in *ISPRS workshop laser scanning*, vol. 38, p. W12, 2011.