

A Upper Limb Rehabilitation System with Motion Intention Detection

Yanyu Su¹, Yan Wu², Yongzhuo Gao¹, Wei Dong¹, Yixuan Sun¹, Zhijiang Du¹

Abstract—Upper limb rehabilitation robotic systems have been studied for decades to improve the rehabilitation training of hemiplegic patients. Ideally such robotic systems should be capable of detecting the intentions of the patients and assisting them as needed. In this paper, we integrate a virtual environment and a low-cost motion sensor, the Microsoft Kinect II, into a robotic upper limb rehabilitation system to detect user motion intentions and generate motion commands for a rehabilitation robot. The system requires users to mimic a pre-programmed bimanual motion sequence shown in the virtual environment. A Gaussian Process based predictive controller uses the motion of the unaffected arm and the programmed motion sequence to estimate the motion intentions of the affected arm. We also adopt this controller in Mirror Therapy, a widely-practised therapeutic intervention method. Two preliminary experiments have been conducted to validate the proposed controller and test proper function and safety of the system including a patient case-study.

I. INTRODUCTION

In this paper, we propose the design of a robotic upper limb rehabilitation system that integrates the Kinect sensor with the use of a virtual reality environment to provide necessary assistance to a user by detecting the motion intention. In this system, we leverage on the convenience provided by Kinect while using a machine learning approach to improve its generalization and robustness in the presence of high sensor noise. The motion sequence prescribed to a given user is demonstrated in the virtual environment by the Virtual Patient. We assume that the user will watch and try to follow the demonstrated preprogrammed motion sequence closely to that of the Virtual Patient and synchronously on both the affected and unaffected arms. Therefore, in place of an exoskeleton/contact sensors for the unaffected arm, we can use the lower-cost Kinect to estimate the time step the unaffected arm of the user is in the prescribed motion sequence, and map that to the required instantaneous motion of the affected arm to generate assistive motions for the system with a user model. There are two benefits of this approach: 1) As the user controls the rehabilitation system, The training programme is controlled as desired; 2) As the motion of the affected arm is generated from that of the unaffected arm, and the range of motion and the length of the arm are similar, the system is unlikely to cost injury to the user.

Several challenges have to be addressed to implement the proposed system: 1) how to build a user model to predict the

user's actual movement in following the prescribed motion sequence; 2) how to estimate which time step the user is moving at; 3) how to generate the prescribed motion sequence for the Virtual Patient and hence the robot. In this work, we propose using Gaussian Process Regression to learn the Patient Model that maps the motion of the Virtual Patient to the user; and estimate the time step by solving an optimal control problem to find the most probable instantaneous motion in the prescribed motion sequence for the user using the Patient Model. As the problem of generating robot motions from captured human motions has been well studied in the literature [1][2], we will adopt a naive kinematics based method to solve this problem.

In order to investigate the scenario of applying the proposed method in a clinical setup, we adopt the method of Mirror Therapy where the user moves both arms as symmetrically as possible while he/she watches the mirror image as if the affected arm behind the mirror is unaffected. Mirror Therapy has been implemented in virtual environment with Kinect in many literatures [3]. However, to our knowledge, none uses robot to help the user in moving the affected arm, which has limited its application. Therefore, adapting the proposed system to the Mirror Therapy will extend its the usability.

The rest of the paper is organised as follows: In Section II, an overview of the full rehabilitation system is presented followed by a detailed description of the predictive controller in Section III. Two experiments are presented and discussed in Section IV before the conclusion for future work.

II. SYSTEM DESCRIPTION

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 Upper Limb Rehabilitation robot (HIT-ULR²), the Kinect, the virtual environment and the Gaussian Process based predictive controller. The system runs in two modes: the training mode and the configuration mode.

In the configuration mode, a therapist can configure the controller to customize the training programs through the Graphical User Interfaces (GUIs) and the Kinect: For example, the therapist may modify the virtual environment through the GUI of Gazebo (the adopted simulator for the virtual environment) and program the training motion sequences using Program by Demonstration (PbD) techniques with the Kinect.

In the training mode, the user wears the rehabilitation robot on the affected arm and observes the virtual environment to carry out training programs. The motion of the

Yanyu Su, Yongzhuo Gao, Wei Dong, Yixuan Sun and Zhijiang Du are with the State Key Laboratory of Robotics and System, Harbin Institute of Technology, China. suyanyu; gaoyongzhuo; sunyixuan; dongwei; duzj01@hit.edu.cn

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

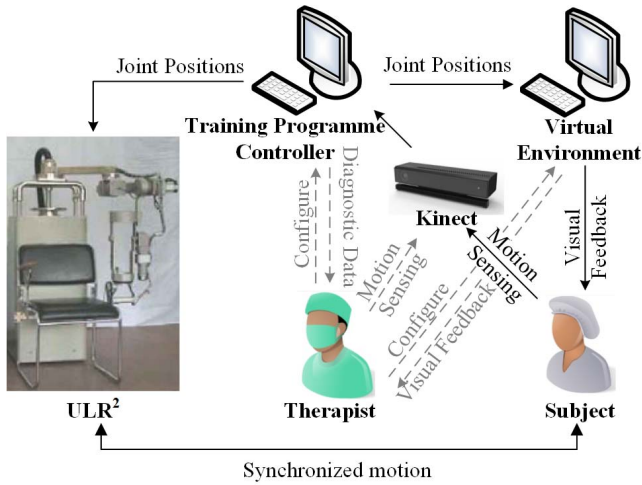


Fig. 1: An overview of the proposed robotic rehabilitation system: Solid lines illustrate data flow of Training Mode while dashed lines show that of the Configuration Mode.

user's unaffected arm is sensed by Kinect. To follow a given 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-ULR². In this paper, we have implemented two training programs: One is the Range of Motion Exercise, where the controller generates the prescribed motion from a preprogrammed sequence; The other one is the Mirroring Exercise, where the controller generates the prescribed motion directly and symmetrically from the unaffected arm.

The Upper Limb Rehabilitation Robot (HIT-ULR²) is a five Degree-of-Freedom (DoF) robotic exoskeleton [4]. A user loosely connects to the exoskeleton with cuffs on the upper arm and on the forearm respectively. An adjusting mechanism is adopted so that the lengths of the exoskeleton segments are adjustable for any individual. Symmetry design is adopted to allow the user to wear the exoskeleton on either arm. The control system of the HIT-ULR² contains a Industrial Personal Computer (IPC) and a Programmable Multi-Axis Controller (PMAC). A position-following controller has been implemented in the control system in our previous work [5].

We implemented a Virtual Patient in the virtual environment with Gazebo [6], which employs physics engines to carry out real-time dynamic simulation of rigid bodies. We mainly adopt it to create, edit and monitor the simulation in an online fashion using its user-friendly GUI, so that any therapist can easily tailor-make a suitable training program and virtual environment. It is worth noting that, the arm of the Virtual Patient is modelled to have the same kinematic structure as that of the HIT-ULR² in order to avoid the correspondence problem.

III. GAUSSIAN PROCESS BASED PREDICTIVE CONTROLLER CONSIDERING MOTION INTENTION DETECTION

The proposed controller, generally follows the Gaussian Process based predictive control framework[7], as shown in Figure 2. It consists of several modules, where four modules shall be emphasized at the beginning: 1) the virtual therapist who represents the therapist carries out accurate movement to illustrate the programmed motion sequence, 2) the virtual patient who represents the patient to act in the virtual environment, 3) the patient model which maps the motion of the virtual therapist to the patient, 4) the transformation which maps the motion of a human to a virtual human and the HIT-ULR². 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

$$P_k = [p_{k_1} \ p_{k_2} \ \dots \ p_{k_{10}}] \in \mathbb{R}^{10 \times 3},$$

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

In order to transfer this information to robot motions, a kinematic model of the skeleton is built as shown in Figure 3(a), where nine frames (from $\{K_1\}$ to $\{K_9\}$) are defined. The origin of each frame is defined at the respective key point. The x axis of the reference frame $\{K_1\}$ is defined by

$$x_{k_1} = n(p_{k_2} - p_{k_6}),$$

with the function $n(\bullet)$ generating the normalized vector, while the y -axis is defined as upward positive and perpendicular to x -axis. The orientation of the shoulder frames, $\{K_2\}$ and $\{K_6\}$, are identical to that of the frame $\{K_1\}$. The y axis of other frames are defined by

$$y_{k_i} = n(p_{k_{i-1}} - p_{k_i}),$$

while the z -axis of other frames are defined by

$$z_{k_i} = n(x_{k_0} \times y_{k_i}),$$

with $i = 2, 3, 4, 6, 7, 8$ as the index of the frames.

2) *Kinematic Model of HIT-ULR²*: The Denavit-Hartenberg (D-H) notation of the HIT-ULR² is given in [4], with which we denote the forward kinematics model as

$$T_{r_i} = f_{r_i}(q),$$

where $i = 1, 2, \dots, 5$ is the index of the frames, $T_{r_i} \in \mathbb{R}^{4 \times 4}$ denotes the transform matrix of frame $\{R_i\}$, $q = [q_1, q_2, \dots, q_5] \in \mathbb{R}^5$ denotes the positions of the joints.

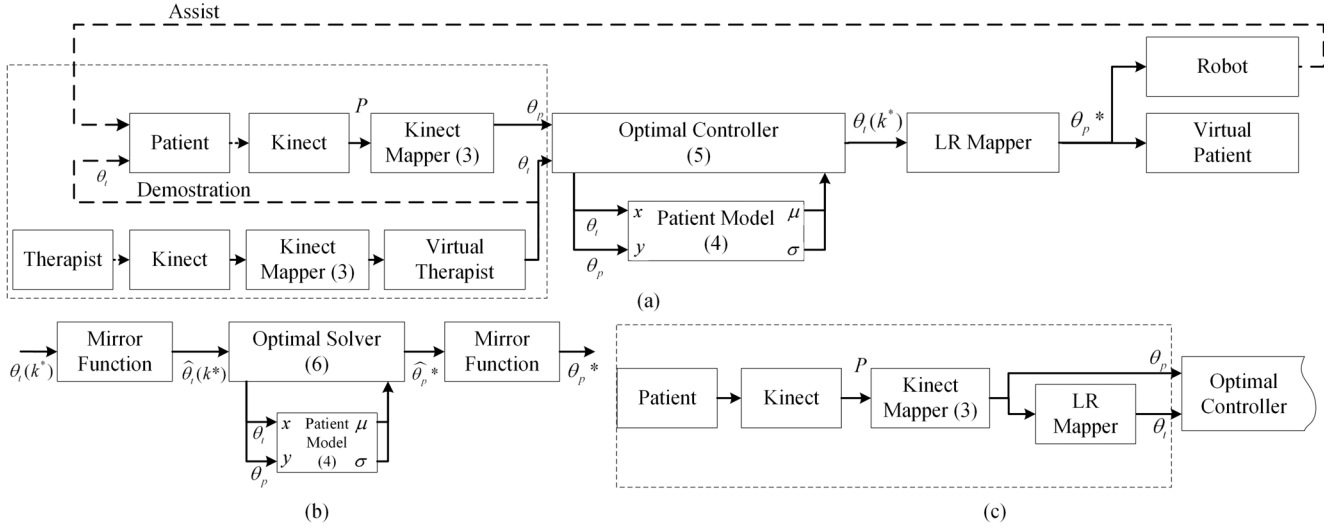


Fig. 2: The block diagram of the controller is shown in (a). The detailed structure of the module LR Mapper is shown in (b). The modification made to the controller for the mirroring training program is shown in (c). Important equations are marked on the respective blocks, for example, referring to (5)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

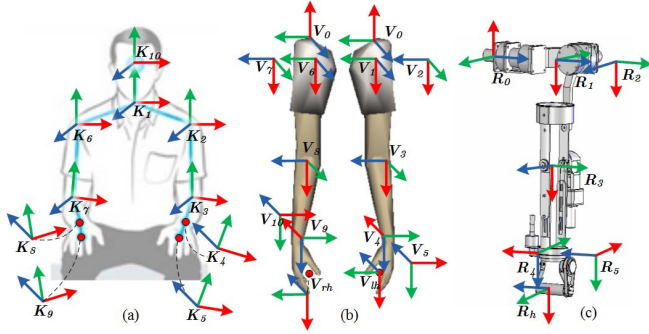


Fig. 3: The kinematic model of the Kinect skeleton, the Virtual Patient and the HIT-ULR² are respectively shown in (a), (b) and (c).

3) *Kinematic Model of Virtual Patient*: The kinematic model of the Virtual Patient is built to be identical to that of the HIT-ULR² as shown in Fig3(b). We denote the forward kinematics as

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

where $i = 1, 2, \dots, 10$ is the index of the frames, $\mathbf{T}_{v_i} \in \mathbb{R}^{4 \times 4}$ denotes the transform matrix of frame $\{V_i\}$, $\boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_{10}] \in \mathbb{R}^{10}$ denotes the joint position of the Virtual Patient.

B. Mapping Kinect Skeleton to HIT-ULR² and Virtual Patient

The mechanical configurations of HIT-ULR² and the Kinect skeleton are different. In this work, we map the Kinect skeleton to the HIT-ULR² by minimizing the error of the directional vectors between the Kinect skeleton and the HIT-ULR² against the joint limits of the HIT-ULR². Therefore, the optimization problem for the left arm can be

defined by

$$\mathbf{q}_k = \underset{\mathbf{q}}{\operatorname{argmin}} \|\mathbf{T}_t * \mathbf{x}_{r_3}(\mathbf{q}) - (-\mathbf{y}_{k_3})\|, \quad (1)$$

such that

$$\begin{aligned} \|\mathbf{T}_t * \mathbf{x}_{r_2}(\mathbf{q}) - (-\mathbf{y}_{k_2})\| &= 0 \\ \|\mathbf{T}_t * \mathbf{x}_{r_{lh}}(\mathbf{q}) - (-\mathbf{y}_{k_5})\| &= 0 \\ (\mathbf{q})_{min} &\leq \mathbf{q} \leq (\mathbf{q})_{max} \end{aligned}$$

and for the right arm

$$\mathbf{q}_k = \underset{\mathbf{q}}{\operatorname{argmin}} \|\mathbf{T}_t * \mathbf{x}_{r_8}(\mathbf{q}) - (-\mathbf{y}_{k_7})\|, \quad (2)$$

such that

$$\begin{aligned} \|\mathbf{T}_t * \mathbf{x}_{r_6}(\mathbf{q}) - (-\mathbf{y}_{k_6})\| &= 0 \\ \|\mathbf{T}_t * \mathbf{x}_{r_{rh}}(\mathbf{q}) - (-\mathbf{y}_{k_9})\| &= 0 \\ (\mathbf{q})_{min} &\leq \mathbf{q} \leq (\mathbf{q})_{max} \end{aligned}$$

where $(\mathbf{q})_{min}$ and $(\mathbf{q})_{max}$ respectively denote the lower and the upper positional limitations of the robot joints, $\mathbf{T}_t \in \mathbb{R}^{4 \times 4}$ is the transform matrix that maps the frame $\{O_0\}$ of the HIT-ULR² to the respective shoulder frame of the Kinect skeleton (the frame $\{K_2\}$ or the frame $\{K_6\}$), \mathbf{x}_{r_2} , \mathbf{x}_{r_3} and \mathbf{x}_{r_h} respectively denote the x -axis of the respective frames of the HIT-ULR².

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

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

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

As the kinematics model of the Virtual Patient is identical to the HIT-ULR², the mapping method for the HIT-ULR² is applicable to mapping the Kinect skeleton to both the left

and right arm of the Virtual Patient using (1) and (2). We denote it as

$$\theta_i = g_{k_i}(\mathbf{P}), \quad (3)$$

where $i = 1, 2, \dots, 10$ is the index of the Virtual Patient joints.

C. User Modelling with Sparse Online Gaussian Process

Our preliminary user study showed that different users may perform a variant of the motion sequence when the Virtual Patient demonstrates the same sequence. In order to describe this phenomenon, we propose to identify the relationship between the motion of the Virtual Patient and the motion of the user by Gaussian Process regression (GPR).

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

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

and its covariance function

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

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 likelihood maximisation. One key drawback of the full GP implementation is the computational cost of the model ($O(n^3)$ for training). Moreover, as readings from any given sensor has perceptual noise, we need a model that is able to learn online so as to generalize and improve on prediction quality as more training samples are provided. In this work, we use the Sparse Online Gaussian Process (SOGP) [8]; an online approximation of the full GP which has a lower computational complexity, $O(|B|)$ where B is the basis vector set (the retained training samples). Note that the maximum size of B can be constrained, effectively rendering the SOGP a constant time algorithm. More detail on the descriptions of the SOGP can be found in [8], [9].

We adopt an SOGP for each joint of the Virtual Patient and describe the motions of the user 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 $i = 1, 2, \dots, 10$ is the index of the joints of the Virtual Patient, x_t is the input, y_t is the output (or target). Our previous work generates training data from a set of calibration postures. In this work, we train the SOGP in an online fashion with the under-taking motion sequences, which avoids the inconsistency of the patient motions due to different motion sequences.

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 of providing the likelihood of a test point once given a likelihood function

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

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

D. Optimal Controller

As shown in Figure 2, given the motion of the patient θ_p and the prescribed sequence by the Virtual Patient $\Theta_t = [\theta_t(1), \theta_t(2), \dots, \theta_t(N)]$ of length $N \in \mathbb{N}$, the optimal controller estimates which time step in the prescribed sequence $k^* \in [1, 2, \dots, N]$ the user has moved to, and outputs the respective movement of the Virtual Patient $\theta_t(k^*)$.

We consider this problem as an optimal control problem. At each control step, the controller should solve the problem to find the time step of the prescribed sequence k^* whose respective instantaneous motion $\theta_p(k^*)$ is closest to the motion of the user θ_p according to the user model defined in (4). Therefore, the optimal control problem is defined by

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

such that

$$I = \begin{cases} \{1, 2, 3, 4, 5\} & \text{with unaffected left arm} \\ \{6, 7, 8, 9, 10\} & \text{with unaffected right arm} \end{cases}$$

$$\theta_{p_i} \in \theta_p$$

$$\theta_{t_i} \in \theta_t(k)$$

$$k \in \{k_c, k_c + 1, \dots, k_c + N_k\}$$

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

E. Generating Motion of Affected Arm for the 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 is not capable of generating motions from an 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

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

where the translation matrix T_a is defined by

$$T_a = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

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

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

such that

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

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 $p_6(\bullet)$ 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 $\hat{\theta}_p^*$ back before sending it to the position controller of the HIT-ULR² and the Virtual Patient

$$\theta_p^* = \begin{cases} \begin{bmatrix} I & 0 \\ 0 & T_a \end{bmatrix} \hat{\theta}_p^*, & \text{when left arm is unaffected} \\ \begin{bmatrix} T_a & 0 \\ 0 & I \end{bmatrix} \hat{\theta}_p^*, & \text{when right arm is unaffected} \end{cases} \quad (7)$$

F. Adapting the Controller to Mirroring Training Program

We simply adopt the proposed controller to mirroring motions by using its own motion as the prescribed motions as shown in Figure 2(c). In this mode, the affected arm is assumed to intentionally perform similar movement to that of the unaffected arm which allows the user to freely manipulate the objects in the virtual environment.

IV. EXPERIMENTS AND RESULTS

Two experiments have been conducted to validate the proposed method for detecting the user's motion intentions (Motion Intention Detection Experiment) and testing the proper function and safety of the system which includes a patient case-study (System Testing Experiment). The virtual environment and the prescribed motion used in the experiments were created and programmed by a skilled therapist. To become familiar with the robot and the experiment task, each subject complete the whole task once before the experiment.

A. Motion Intention Detection Experiment

In this experiment, ten healthy subjects (five males and five females) were asked to mimic three prescribed motions, shown in Figure 4, without wearing the robot to investigate how accurate the controller could detect the user's motion

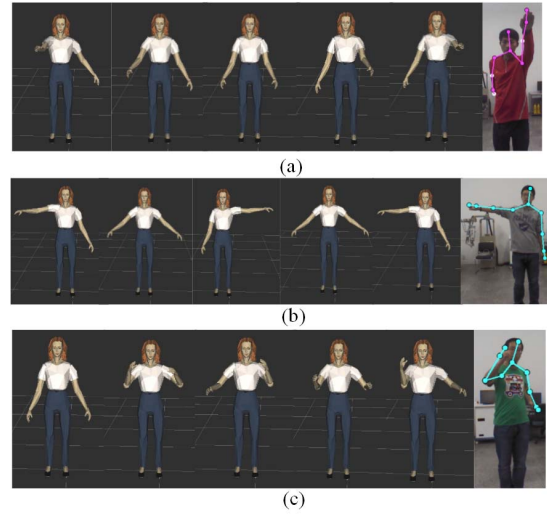


Fig. 4: Three motions (shown by Virtual Patient) were used in motion intention detection experiment. Subjects were required to repeat each motion for eight times in a trial.

intentions. Thirty trials were taken in the experiment. In each trial, the subject was required to repeat each motion for eight times. The data of first four times was used to generate the training data set of the patient model. The joint positions of the subject, the joint positions of the Virtual Patient and the generated motion of the Virtual Patient calculated by (7) were recorded synchronously to estimate the performance.

Without the loss of generality, we assumed that the left arm of the subjects were affected to configure the controller. 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 control strategy.

We define the error of the generated motion, that is, the maximum error between the joint positions of the user and the joint positions of the Virtual Patient generated by the proposed controller

$$m_p = \max_{i=1,2,3} |\theta_{p_i}^* - \theta_{p_i}|, \quad (8)$$

where θ_{p_i} is the Virtual Patient joint positions directly mapped from the actual human motion with (3). The wrist joints are not considered here due to the noise of the Kinect. The lower error means the generated motion of the Virtual Patient is more similar to the actual motion of the subject.

We also define the error of the pre-programmed motion, that is, the maximum error between the pre-programmed joint positions of the user and the joint positions of the Virtual Patient

$$m_c = \max_{i=1,2,3} |\theta_{t_i} - \theta_{p_i}|. \quad (9)$$

The pre-programmed motion is commonly used as the reference to a position controller or an impedance controller in many robotic rehabilitation systems that are not adaptive to the user at the motion level but usually at the task level [10], [11], [12]. Therefore, we use this metric as a compare to the proposed control strategy. It is worth noting that since

many robotic rehabilitation systems use the position based impedance controller to adapt to the patient at the motion level [13], and the proposed controller output positions as the command, the proposed controller is applicable to the impedance control strategies.

An trial example is shown in Figure 5. Short delays and rhythmic mismatches of the subject can be observed in Figure 5(a). This is common when a subject is learning a movement. And it is also observable that the generated motion generally matches the ground truth in temporal space, so the proposed controller is able to adapt to the rhythm of the subject.

The error of the generated motion is less than that of the pre-programmed motion as shown in Figure 5(b). The average error of the generated motion $3.4 \pm 2.9^\circ$ is less than that of the pre-programmed motion $16.0 \pm 12.3^\circ$ (mean and standard deviation). The t-test on the result of the trail shows that the error of the generated motion is less than 5° (at the 5% significance, $p < 1.0 \times 10^{-5}$), while the error of the pre-programmed motion is not less than 14° ($p = 0.97$).

As shown in Figure 6, the average error of the generated motion for thirty trials is $4.2 \pm 5.7^\circ$ while that of the pre-programmed motion is $14.2 \pm 12.2^\circ$. The t-test on the results of all the trails shows that the error of the generated motion is less than that of the pre-programmed motion ($p < 1.0 \times 10^{-5}$). It also shows that the error of the generated motion is less than 5° ($p < 1.0 \times 10^{-5}$), while the error of the pre-programmed motion is not less than 14° ($p = 0.48$).

The best aiming angle (a kinematic index to evaluate the movement accuracy of a user) of average users after rehabilitation therapy is $14.26 \pm 10.25^\circ$ [14]. Therefore, the error of 5° can be considered within the error-bound of the requirement for a user. Meanwhile, adopting the accuracy requirement proposed in [3], the error of 5° can be considered acceptable for clinical purposes because visual observation error in a clinical setting has been shown to be approximately 10% of the movement range [15] as the movement range is around 90° in the experiment.

B. System Testing Experiment

In this experiment, 10 healthy subjects and 1 hemiplegic patient (male, age 34, in acute phase, no spasticity, no motion ability, affected left arm and leg, unaffected right arm and leg, no cognitive and communication disorder) were asked to performed the pick-and-place task in an virtual environment to test proper function and safety of the system. The subjects and the patient were asked to mimic a calibration motion (individually moving his/her arm covering the range of motion $0 - 90^\circ$) before each trial to generate the training data set of the patient model. The controller was running in the mirroring training mode, so the subjects could freely interact with the virtual environment.

All the healthy subjects and the patient were able to complete the required task using the proposed system. An example of a subject conducting the task is shown in Figure 7. The motion of the robot and that of the unaffected arm of the subject were recorded to evaluate how the robot followed the subject and how the subject exercised in the experiment.

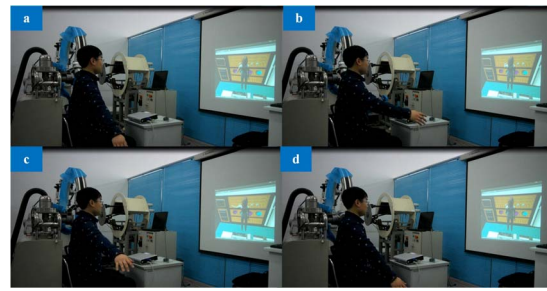


Fig. 7: A healthy subject is conducting the pick-and-place task in the virtual kitchen. He moves toward the object as shown in (a) before picking the object shown in (b) and puts the object in to the bowl as shown in (c) and (d).

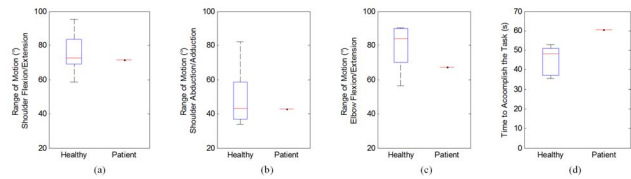


Fig. 8: Box plot of the used range of motion and the used time to accomplish the task of the healthy subjects and the patient.

The used range of motion is a common index to evaluate which joints were mainly involved to solve the task [16]. And the time to accomplish the task is also a common index to evaluate how the user could accomplish the task. The result statistics is shown in Figure 8 in box plot. It is observable that the range of motion with respect to the shoulder joint of the patient is similar to that of the healthy subjects. The range of motion with respect to the elbow joint of the patient is lower than most of the healthy subjects but still inside the range of all the healthy subjects. The time for the patient to accomplish the task is higher, which is understandable as his motion is slower than that of the healthy subjects.

V. CONCLUSION

In this paper, we presented a upper limb rehabilitation robotic system integrated with virtual environment and motion sensing. We contributed a method to address the problem of detecting the user's motion intention. We focused on the training program where the user follows a motion demonstrated by the therapist. With the assumption that the user would intentionally try to synchronously move both the unaffected arm and the affected arm to follow the prescribed motion, we proposed and implemented a Gaussian Process based model predictive controller that predicts the motion intention of the affected arm based on the motion of the unaffected arm. Our experimental results have shown that the proposed controller could predict the motion intentions of the user with the prediction error less than 5° on average which meets the requirements for therapy.

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

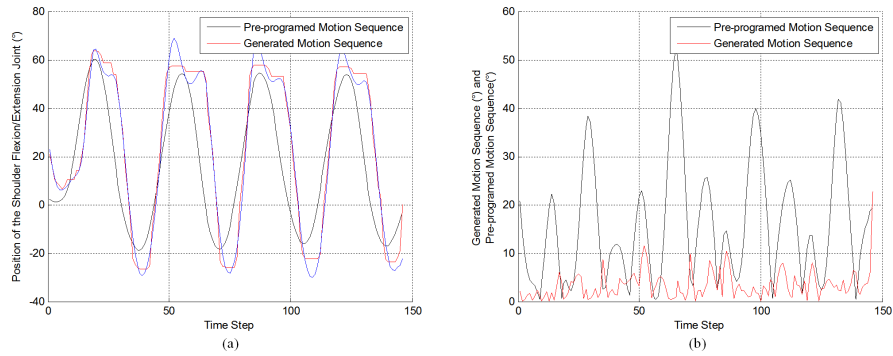


Fig. 5: An example of the pre-programmed motion, the generated motion of the proposed controller and the ground truth is shown in (a) with the error of the generated motion and the error of the pre-programmed motion is shown in (b).

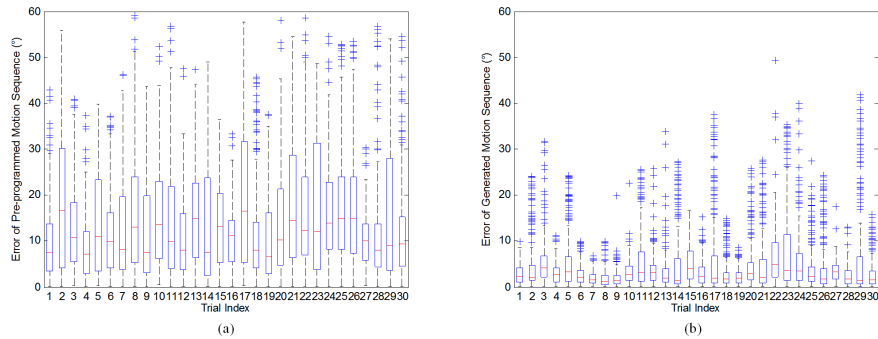


Fig. 6: Box plot of the error of pre-programmed motion (a) and the error of generated motion of the proposed controller (b).

into the system to build a force feed-back training system. Another emergent improvement is to evaluate the system with more hemiplegic subjects. We will conduct clinical experiments on the proposed to investigate how the system would work with stroke patients.

REFERENCES

- [1] 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.
- [2] 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.
- [3] 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, pp. 1–10, 2014.
- [4] 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. 6840–3, jan 2005.
- [5] 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.
- [6] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 3, pp. 2149–2154, IEEE, 2004.
- [7] B. Likar and J. Kocijan, "Predictive control of a gas-liquid separation plant based on a Gaussian process model," *Computers & chemical engineering*, vol. 31, no. 3, pp. 142–152, 2007.
- [8] L. Csató and M. Opper, "Sparse on-line gaussian processes," *Neural computation*, vol. 14, pp. 641–68, mar 2002.
- [9] H. Soh, Y. Su, and Y. Demiris, "Online Spatio-Temporal Gaussian Process Experts with Application to Tactile Classification," in *2012 IEEE International Conference on Intelligent Robots and Systems*, (Vilamoura, Portugal), pp. 4489–4496, IEEE, 2012.
- [10] H. I. Krebs, B. T. Volpe, M. L. Aisen, and N. Hogan, "Increasing productivity and quality of care: robot-aided neuro-rehabilitation," *Journal of rehabilitation research and development*, vol. 37, no. 6, pp. 639–652, 2000.
- [11] P. S. Lum, C. G. Burgar, M. der Loos, P. C. Shor, M. Majmundar, and R. Yap, "MIME robotic device for upper-limb neurorehabilitation in subacute stroke subjects: A follow-up study," *Journal of rehabilitation research and development*, vol. 43, no. 5, p. 631, 2006.
- [12] G. Xu, A. Song, and H. Li, "Adaptive impedance control for upper-limb rehabilitation robot using evolutionary dynamic recurrent fuzzy neural network," *Journal of Intelligent & Robotic Systems*, vol. 62, no. 3-4, pp. 501–525, 2011.
- [13] L. Marchal-Crespo and D. J. Reinkensmeyer, "Review of control strategies for robotic movement training after neurologic injury," *Journal of neuroengineering and rehabilitation*, vol. 6, no. 1, p. 20, 2009.
- [14] L. Zollo, L. Rossini, M. Bravi, G. Magrone, S. Sterzi, and E. Guglielmelli, "Quantitative evaluation of upper-limb motor control in robot-aided rehabilitation," *Medical & biological engineering & computing*, vol. 49, no. 10, pp. 1131–1144, 2011.
- [15] H. L. Hurkmans, J. B. Bussmann, and E. Benda, "Validity and interobserver reliability of visual observation to assess partial weight-bearing," *Archives of physical medicine and rehabilitation*, vol. 90, no. 2, pp. 309–313, 2009.
- [16] M. Guidali, A. Duschau-Wicke, S. Broggi, V. Klamroth-Marganska, T. Nef, and R. Riener, "A robotic system to train activities of daily living in a virtual environment," *Medical & biological engineering & computing*, vol. 49, no. 10, pp. 1213–1223, 2011.