2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007

Intelligent Control Framework for Robotic Rehabilitation after Stroke

Duygun Erol & Nilanjan Sarkar, *Senior Member, IEEE*

Abstract — **This paper presents a new approach to robot assisted rehabilitation for stroke patients. The control architecture is represented in terms of hybrid system model combining a high-level and a low-level controller. The main focus of this paper is to present an intelligent controller, which is the high-level controller in the control architecture. The high-level controller is designed to monitor the progress and safety of the rehabilitation task. It also makes decisions on the modification of the task that might be needed for the therapy. Experimental results on unimpaired subjects are presented to demonstrate the efficacy of the high-level controller.**

Keywords: robot-assisted rehabilitation, intelligent controller, movement tracking training

I. Introduction

Stroke is a highly prevalent condition [1], especially among the elderly, that results in high costs to the individual and society [2]. According to the American Heart Association (2006), in the U.S., approximately 700,000 people suffer a first or recurrent stroke each year [1]. It is a leading cause of disability, commonly involving deficits of motor function. Recent clinical results have indicated that movement assisted therapy can have a significant beneficial impact on a large segment of the population affected by stroke or other motor deficit disorders. In the last few years, robot-assisted rehabilitation for rehabilitation of the stroke patients has been an active research area, which provide repetitive movement exercise and standardized delivery of therapy with the potential of enhancing quantification of the therapeutic process [3]-[6].

The first robotic assistive device used as a therapeutic tool is the MIT-MANUS, which uses an impedance controller to provide assistance to the patient [5]. Mirror Image Movement Enabler (MIME) [4] uses a PUMA 560 manipulator to provide assistance to move the subject's arm with a pre-programmed position trajectory using Proportional-Integral-Derivative (PID) controller. Assisted Rehabilitation and Measurement (ARM) Guide [3] is another robotic system that is capable of generating both horizontal and vertical motion, which provides assistance or resistance to the patient's movement to complete the reaching task. The GENTLE/s is used to provide assistance to patients to move to the target positions along with a predefined path using admittance control. The subject's movement trajectory is represented in the virtual environment [6]. Studies with these robotic devices verified that robot-assisted rehabilitation results in improved performance of functional tasks.

The existing robotic rehabilitation systems mainly use low-level controllers to assist the movement of patients' arms. In these cases, the task parameters are pre-defined at the beginning of the therapy. The therapist continuously monitors the patient's progress and quickly determines if the task parameters are needed to be changed to adapt to the patients' performance. Then these task parameters are updated by the therapist to be executed by the low-level controllers. In addition, the therapist is required to pay attention to the safety related issues during the robot assisted therapy. If such an issue occurs, the therapist needs to take necessary action to ensure the safety of the patients.

In this work we develop an intelligent control framework to help the therapist in i) determining the task parameters dynamically based on patients' performance and implementing the new set of parameters; and ii) monitoring the safety related events in an automated manner and generating an accommodating plan of action should such an event happen. We believe that such an intelligent controller will likely to help in therapist's decision-making and make the robotic rehabilitation process safer. The primary focus of this paper is to present an intelligent controller, which is called a high-level controller, to monitor the progress of the task and make decisions on the modification of the task that might be needed for the therapy. This high-level controller works in coordination with low-level controllers.

This paper is organized as follows. It first presents the overall control architecture in Section II. Then the rehabilitation robotic system is presented in Section III. The high-level controller of the overall control architecture has been described in Section IV. Results of the experiments are presented in Section V to demonstrate the efficacy of the high-level controller on unimpaired subjects. Section VI discusses potential contributions of this work and possible directions for future work.

II. Control Architecture

Let us first present a broad overview of the control architecture of robot-assisted rehabilitation that we present in this paper (Fig. 1). This architecture is represented by hybrid system model. A hybrid system model has three parts, a "Plant", a "Controller" and an Interface [7],[8]. In order to avoid confusion about terminology, we call the controller in hybrid system model a high-level controller in this paper. The continuous part, identified as the "Plant" is

D. Erol is with the Department of Electrical Engineering and Computer Science , Vanderbilt University, Nashville, TN, 37235 (e-mail: duygun.erol@vanderbilt.edu).

N. Sarkar is with the Department of Electrical Engineering and Computer Science and Department of Mechanical Engineering, Vanderbilt University, Nashville, TN, 37235 (e-mail: nilanjan.sarkar@vanderbilt.edu).

1-4244-0602-1/07/$20.00 ©2007 IEEE.

typically described by differential/difference equations. In this architecture, the low-level assistive controller represents the "Plant", which provides robotic assistance to subjects to complete a rehabilitation task. The design of the low-level assistive controller has been described in more details in [9]. The high-level controller includes a discrete decision process that is typically a discrete-event system (DES) described by a finite state automaton. The proposed high-level controller makes decision about the task that is required for the therapy during the robotic assistance (Fig 1). The high-level controller and the plant cannot communicate directly in a hybrid control system because each utilizes different types of signals. Thus an interface is required which can convert continuous-time signals to sequences of symbols and vice versa. There has been no work to our knowledge on designing such a hybrid system for rehabilitation purposes. However, there has been some work on developing such hybrid controllers in other fields, such as industrial robotics, medicine, and manufacturing [10].

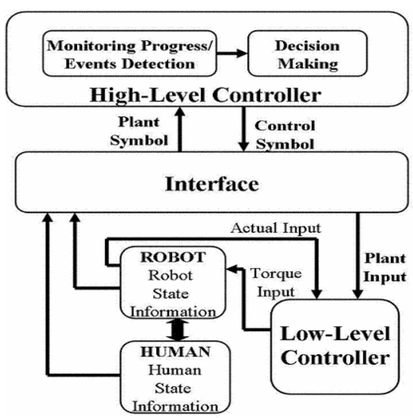


Figure 1: Control Architecture

In order for the high-level controller to decide the necessary control actions, the state information from the robot and the human is observed by the process monitoring module through the interface (Fig. 1). The interface triggers events pertinent to the task and communicates to the process monitoring block of the high-level controller. Once these events are triggered, the decision making module of the high-level controller determines what actions to be taken in response to these events. The high-level control's instructions are sent to the low-level assistive controller through the interface, which then executes these actions. The proposed architecture is flexible and extendible in the sense that new events can be included and detected by simply monitoring the additional state information from the human and the robot, and accommodated by introducing new low-level assistive controllers.

III. Rehabilitation Robotic Device

The proposed rehabilitation robotic system uses a PUMA 560 robotic manipulator, which is augmented with a force-torque sensor and a hand attachment device (Fig. 2). The microcontroller board of the PUMA is replaced to develop an open architecture system to allow implementing the low-level assistive [9] and high-level controllers. We have introduced a tool frame to include the location of the human arm through the hand attachment device.

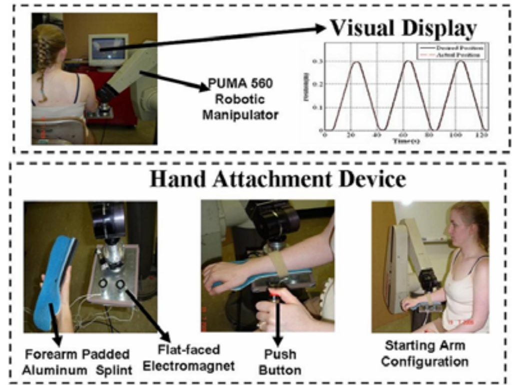


Fig. 2. Subject Arm attached to Robot

In order to record the force and torque, an ATI Gamma force/torque sensor is used. The robot is interfaced with Matlab/Real-time Workshop to allow fast and easy system development. The force values recorded from the force/torque sensor are obtained using a National Instruments PCI-6031E data acquisition card with a sampling time of 0.001 seconds. The joint angles of the robot are measured using encoder with a sample time of 0.001 seconds from a Measurement Computing PCI-QUAD04 card. The torque output to the robot is provided by a Measurement Computing PCIM-DDA06/16 card with the same sample time. A computer monitor is placed in front of the subject to provide visual feedback about his/her motion trajectory during the execution of the task.

Since in this work we are primarily interested in effecting assistance to the upper arm, a hand attachment device is designed where the subject's arm is strapped into a splint that restricts wrist and hand movement. The subject is asked to use the arm to make a movement with or without the assistance from the robot [3]-[6]. Forearm padded aluminum splint (from MooreMedical), which ensures the subject's comfort, is used as a splint in this device. A steel plate with proper grooves is designed that holds two small flat-faced electromagnets (from Magnetool Inc.) that are screwed on it. This plate is also screwed with the force-torque sensor, which provides a rigid connection with the robot. A light-weight steel plate under the splint is attached, which is then attached to the electromagnets of the plate. These electromagnets are rated for continuous duty cycle (100% duty cycle), i.e., they can run continuously at normal room temperature. Pull ratings of these magnets are 40lb. Two electromagnets have been used to have a larger pulling force to keep the splint attached to the hand attachment device. An automatic release (AU) rectifier controller (Magnetool Inc.) has been used to provide a quick release of these electromagnets. A push button, which has been connected to the AU Rectifier Controller, is used to magnetize and demagnetize the electromagnets when the subject wants to remove the hand attachment device from the robotic manipulator in a safe and quick manner.

Ensuring safety of the subject is a very important issue when designing a rehabilitation robotic system. Thus, in case of emergency situations, therapist can press an emergency button. The patient and/or the therapist can quickly release the patient's arm from the PUMA 560 by using the quick-release hand attachment device to deal with any physical safety related events. In order to release the subject's arm from the robot, the push button is used. When the push button is pressed electromagnets are demagnetized instantaneously and the subject is free to remove the splint from the robot. The push button can also be operated by a therapist.

IV. High-Level Controller

The high-level controller monitors the progress of the task, the status of the plant, and makes decision on the modification of the task that might be needed for the therapy. The high-level controller decisions are executed by the low-level assistive controller to accomplish the task requirements. In this section, we first present the theory of the high-level controller. The design of the high-level controller needs to consider the specific task that it needs to monitor. Thus we present the task description, and then provide the details of the design of the high-level control.

A. Theory

The high-level controller is a discrete-event system (DES) deterministic finite automaton, which is specified by $D = (\widetilde{P}, \widetilde{X}, \widetilde{R}, \psi, \lambda)$ [7],[8]. Here $\widetilde{P}$ is the set of discrete states. Each event is represented as a plant symbol, where $\widetilde{X}$ is the set of such symbols, for each discrete state. The next discrete state is activated based on the current discrete state and the associated plant symbol using the following transition function: $\psi : \widetilde{P} \times \widetilde{X} \rightarrow \widetilde{P}$. In order to notify the low-level assistive controller the next course of action in the new discrete state, the controller generates a set of symbols, called control symbols denoted by $\widetilde{R}$ using an output function: $\lambda : \widetilde{P} \rightarrow \widetilde{R}$. The action of the high-level control is described by the following equations:

$$\widetilde{p}_j[n] = \psi(\widetilde{p}_i[n-1], \widetilde{x}_k[n]) \tag{1a}$$

$$\widetilde{r}_c[n] = \lambda(\widetilde{p}_j[n]) \tag{1b}$$

where $\widetilde{p}_i, \widetilde{p}_j \in \widetilde{P}$, $\widetilde{x}_k \in \widetilde{X}$ and $\widetilde{r}_c \in \widetilde{R}$. i and j represent the index of discrete states. k and c represent the index of plant symbols and control symbols, respectively. n is the time index that specifies the order of the symbols in the sequence.

B. Task Description

We choose a reaching task that is commonly used to increase the active range of motion (AROM) in shoulder and elbow in preparation of later functional reaching activities in rehabilitation of upper extremity after stroke. In this task, the subjects are asked to move their arms in the forward direction to reach a desired point in space and then bring it back to the starting position in a repeated manner. We ask the subjects to follow a visually presented desired motion trajectory which is likely to command their concentration. It had been shown in the literature that the movement tracking task that required cognitive processing achieved greater gains in performance than that of movement training that did not require cognitive processing [11]. In this work, we constrain the motion of the arm in the horizontal plane and in one direction (along the Y-axis). The idea here is to improve the ability of subject's arm movement in one direction at a time by helping him/her to improve his/her speed of movement, which is an important criterion to measure the success of a therapy. The tip of the position trajectory that the subject is required to follow represents the velocity of the task trajectory. During the execution of the reaching task, the number of times subject needed robotic assistance to track the desired motion is recorded. Task parameters are modified based on this number to make the task more or less challenging using the high-level control. Safety related events are monitored during the task execution so that the high-level controller can make decision for the next plan of action to ensure safety of the subjects.

C. High-Level Controller Design

The high-level controller first detects the state information from the robot and the human through the interface, and then determines the actions to be taken in response to this information. The state information from the robot and the human can be a continuos signal or a discrete value. Let S_{Rn} and S_{Hn} represent the sets of robot and human state information, respectively. In this research, the continuos signals that are detected from the robot are: i) robot's joint angles (S_{R1}), ii) the force reference calculated for the planner given in [9] (S_{R2}), iii) the subject's velocity, which is measured from the tool frame velocity (S_{R3}). The discrete value detected from the robot is the subject's progress during the tracking task (S_{R4}). In order to find S_{R4}, the number of times subject needed robotic assistance at 10th trial (n_{10}) and at 50th trial (n_{50}) were recorded. Decision logic is defined to determine the value of S_{R4}:

$$\text{if} \quad n_{50} < \left(n_{10} - \left(n_{10} * \frac{\Delta p}{100}\right)\right) \text{ then } \{S_{R4}=1\}, \tag{2a}$$

$$\text{elseif} \quad n_{50} > \left(n_{10} + \left(n_{10} * \frac{\Delta p}{100}\right)\right) \text{ then } \{S_{R4}=-1\}, \tag{2b}$$

$$\text{else} \{S_{R4}=0\} \tag{2c}$$

Δp is the percentage change of velocity that can be defined by the therapist and it can be person specific. We detect two discrete signals from the human state information, which can take only 0 or 1 value: i) when the pause or stop button is pressed (S_{H1}), and ii) when the pause button is released (S_{H2}). However, continuos signals such as heart rate detection of patients using electrocardiogram (ECG) signal can also be included as one of the human state information in the control architecture. In this task, we define the following plant states $\widetilde{p}$: *stay, difficult, easy, stop* and *pause*. *Stay* ($\widetilde{p}_1$) implies the subject needs to continue the task at the same difficulty level. *Difficult* ($\widetilde{p}_2$) means the subject has improved his/her task performance and task parameters need to be more challenging. Similarly, *easy* ($\widetilde{p}_3$) implies changing the task parameters to make the task easier. *Stop* ($\widetilde{p}_4$) and *pause* ($\widetilde{p}_5$) are defined in their usual ways.

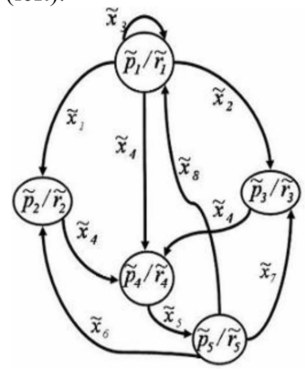
Robot and human state information is monitored to trigger relevant events to modify the task. When these events are triggered, the interface provides the necessary plant symbol ($\tilde{x}$) to the high-level controller. Currently we have defined nine events for the proposed high-level controller. However, the number of events can be easily extended. Five of these events (*E1, E2*, *E3, E4* and *E5*) are robot generated, and three of these events (*E6, E7* and *E8)* are human generated events. The other event, which is a secondary event, is called *SE1*. This is used to detect the previous state when the subject wants to continue with the task after he/she stops. The high-level controller needs to know which state is active before the pause or stop button is pressed in order to provide the same task parameters to the subject when he/she resumes the task. For example, when the subject presses pause button, a value is assigned to *SE1*. This value is retrieved when the subject resumes the task so that he or she can continue the therapy with the same task requirements. Events are reset at the beginning of task execution. Additionally, the triggered event is reset when a new event occurs. When the subject requires less, more or same level of robotic assistance to track the desired trajectory, *E1*, *E2* and *E3* is triggered, respectively. *E4* occurs when the robot's joint angles are out of range. If the force reference provided to the low-level assistive controller to assist the subject and the subject's velocity are above predefined threshold values, then *E5* and *E6* are triggered, respectively. *E7* occurs when the subject presses the pause or the stop button. In order to continue with the task, the subject resets the pause button and *E8* event is triggered. Plant symbols ($\tilde{x}$) are designed based on the events (Table I). The *joint_limits* are known from the robot's specifications. $F_{dthreshold}$ and $\dot{x}_{threshold}$ are determined by the therapist at the beginning of the task execution. Note that if any of *E4*, *E5*, *E6*, and *E7* or their combinations occurs then the state *stop* is activated. Thus we assign the same plant symbol, $\tilde{x}_4$ for these events.

TABLE I: Plant Symbols for the High-Level Controller

Signals from Human and Robot	Event Triggered	Plant Symbol
$S_{R4}=1$	$E1=1$	$\tilde{x}_1$
$S_{R4}=-1$	$E2=1$	$\tilde{x}_2$
$S_{R4}=0$	$E3=1$	$\tilde{x}_3$
$S_{R1}>$ *joint_limits or*	$E4=1$	$\tilde{x}_4$
$S_{R2}>F_{dthreshold}$ *or*	$E5=1$	
$S_{R3}>\dot{x}_{threshold}$ *or*	$E6=1$	
$S_{H1}=1$	$E7=1$	
$S_{H2}=1$	$E8=1$	$\tilde{x}_5$

The secondary event, *SE1*, is defined as follows: if the state is *difficult* and *E7*=1, then *SE1*=1. We assign a corresponding plant symbol $\tilde{x}_6$. Similarly, if the state is *easy* and *E7*=1, then *SE1*=2, and the plant symbol $\tilde{x}_7$ is assigned. If the state is *stay* and *E7*=1, then *SE1*=3. We assign a corresponding plant symbol $\tilde{x}_8$. *SE1* releases state information when *E7*=0 and *E8*=1.

When any of these events is triggered, then the high-level controller decides the next plan of action to modify the task. When an event is triggered, the corresponding plant symbol ($\tilde{x}$) is generated by the interface. The current state ($\tilde{p}$) and the plant symbol ($\tilde{x}$) are used by the high-level controller to determine the next state. Then the high-level controller generates the corresponding control symbol ($\tilde{r}$) for this new state and provides it to the interface. The control mechanism of the proposed high-level controller is shown in Fig. 3 (left).



Control Symbol	Constant Value
$\tilde{r}_1$	$\alpha=0,\beta=1$
$\tilde{r}_2$	$\alpha=1,\beta=1$
$\tilde{r}_3$	$\alpha=-1,\beta=1$
$\tilde{r}_4$	$\beta=0$
$\tilde{r}_5$	$\beta=0$

Figure 3: Controller for the Rehabilitation System

In this figure, $\tilde{r}_c$ s are corresponding control symbols for each plant symbol $\tilde{x}_k$, where *k=1,2,...8*, *c=1,2,...5*. Any event that generates corresponding plant symbols $\tilde{x}_k$ along with the current state information $\tilde{p}_i$ determines the next $\tilde{p}_j$ and as a result, $\tilde{r}_c$, where *i=1,2,...5* and *j=1,2,...5*. In our application only one state is active at any given time, and therefore we uniquely assign a control symbol $\tilde{r}_i$ for each state $\tilde{p}_i$. Since the low-level assistive controller cannot interpret the control symbols, the interface converts them to the appropriate values for α and β for (3) to execute the task. The available control symbols $\tilde{r}_i$ and their corresponding α and β values for the plant input are defined in a table in Fig. 3 (right). The plant equation which determines the desired velocity for the low-level assistive controller is defined as:

$$\dot{x}_{dm} = \beta(\dot{x}_d + (\alpha * delta)) \quad (3)$$

where *delta* is selected as a constant value to increase and decrease the $\dot{x}_d$, which makes the task more or less challenging. $\dot{x}_{dm}$ is the new desired velocity value used to determine the new $\dot{x}_u$ and $\dot{x}_l$. The $\dot{x}_{ave}, \dot{x}_{uave}$ and $\dot{x}_{lave}$ were calculated and if $\dot{x}_{lave} < \dot{x}_{ave} < \dot{x}_{uave}$ was not satisfied then the low-level assistive controller was activated to provide assistance to complement subject's effort to complete the task in a precise manner [9].The Matlab/Simulink/Stateflow software was used to implement the proposed high-level controller [12].

V. RESULTS

In this section we present the experimental procedure and the results of the experiments with unimpaired subjects. Three female and one male subjects within the age range of 25-30 years, right-handed subjects took part in the experiment. The subjects tried to track the desired position trajectory by visually looking at the computer screen. $\dot{x}$ was selected as *0.02m/s*, which was chosen in consultation with a physical therapist. The $\dot{x}_u$ and $\dot{x}_l$ were selected as *25%* more and less of $\dot{x}$, which were *0.025m/s* and *0.015m/s*, respectively. The condition $\dot{x}_{lave} < \dot{x}_{ave} < \dot{x}_{uave}$ was checked to decide the low-level assistive controller activation. Subjects were asked to execute the tracking task 50 times. The number of trials and the number of times subject needed robotic assistance were recorded (Table II). Friction and gravity compensation were always activated in order for the subject to move the robot along with his/her arm in an effortless way.

TABLE II: Number of Times Robot Assisted

Trial Range	1-10	11-20	21-30	31-40	41-50
Assistance for P1	8	6	5	4	3
Assistance for P2	14	13	13	12	11
Assistance for P3	13	11	9	8	7
Assistance for P4	12	12	11	10	9

In order to demonstrate the efficacy of the proposed high-level controller, we had designed two experiments. In the first experiment, we had demonstrated the efficacy the proposed high-level control to modify the task when the subject improved his/her movement ability to track the desired trajectory. In the second experiment, we had demonstrated the efficacy of the high-level controller to modify the task in order to ensure the safety of the subjects.

In the first experiment, we had used P1's low-level assistive controller results. Δp was selected as *30*, which could be varied based on subject's progress and therapist's choice. It was observed from Table II that $n_{10}=8$ and $n_{50}=3$ and Equation (2a) was satisfied, thus *E1* was triggered and the plant symbol $\tilde{x}_1$ was generated from the interface. *difficult* ($\tilde{p}_2$) state became active and the control symbol $\tilde{r}_2$ was generated. The interface converted this control symbol to $\alpha=1$ and $\beta=1$. Amount of the increment (*delta*) to increase the difficulty level of the task was an important issue that needed to be decided. In rehabilitation therapies, increasing $\dot{x}_d$ with a small increment would be more desirable especially for low-functioning stroke patients. In this experiment, we had incremented $\dot{x}_d$ by *20%*, where *delta=0.004.* New desired velocity was calculated using (3), which was *0.024m/s*. The velocity boundaries were calculated as *0.03m/s* and *0.018m/s* for $\dot{x}_u$ and $\dot{x}_l$, respectively. We had asked P1 to perform the tracking task 50 times with this new velocity boundary. Low-level assistive controller assisted the subject as and when they were out of the new velocity band. It was observed that the P1 needed more robotic assistance when the desired velocity to complete the task was increased. It could be seen that P1 learned how to accomplish the task with practice (Table III).

TABLE III: Number of Times Robot Assisted for P1 with New Velocity

Trial Range	1-10	11-20	21-30	31-40	41-50
Assistance for P1	11	10	9	8	7

In the second experiment, we had assumed a safety event had occurred when P1 was performing the task with new increased velocity band. P1 did not feel comfortable and wanted to stop for a while and then reset the pause button in order to complete the rest of the task. When the task started *E1* was triggered and the plant symbol $\tilde{x}_1$ was generated from the interface. *difficult* ($\tilde{p}_2$) state became active and the control symbol $\tilde{r}_2$ was given to the interface. The interface converted this control symbol to constant values $\alpha=1$ and $\beta=1$. The plant equation (3) was used to calculate $\dot{x}_{dm}$ (the desired velocity), which was *0.024m/s*. The reaching task required subject to move *0.3m* and then came back to the starting position. Thus, the initial position (*0*), desired position (*0.3*) and desired $\dot{x}_{dm}$(*0.024m/s*) was provided to the reference block to generate the smooth desired velocity trajectory from *A* to *B* (Fig. 4).

When P1 pressed the pause button at *B*, *E7* was triggered. When *E7* was triggered, plant symbol $\tilde{x}_4$ was generated from the interface and *stop* ($\tilde{p}_4$) state became active. When *stop* state was active, the high-level control provided the control symbol $\tilde{r}_4$ and $\beta=0$ was given to (3) and $\dot{x}_{dm}$ was determined as zero. The zero velocity could cause sudden stop. In order to prevent P1 from sudden stop, the reference generator block was used to provide a smooth velocity trajectory to bring the motion to stop. In this case, the velocity was detected when *E7* was triggered and the desired velocity was given as zero and using the reference generator block, the smooth desired velocity was given to the low-level assistive controller from *B* to *C* (Fig. 4). It could be seen that P1's position did not change after the velocity became zero until P1 reset the pause button. *SE1* was set to *1* because the state was *difficult* and *E7*=1. When the subject reset the pause button, *E8* was triggered and $\tilde{x}_5$ plant symbol was given to the interface, and *pause* ($\tilde{p}_5$) state became active and the high-level controller provided $\tilde{r}_5$. Then $\tilde{x}_6$ was given to the interface because *SE=1*. The corresponding control symbol $\tilde{r}_2$ was generated, and $\alpha=1$ and $\beta=1$ values were given to (3) to calculate $\dot{x}_{dm}$, which was *0.024m/s*. It could be seen that the high-level controller resumed the task in such a manner that the subject could continue with the therapy with the same task parameters. The subject's position at the time of the triggering of *E8* was detected and was given as an initial position to the reference generator block; the desired position was set to *0.3*. The velocity trajectory from *C* to *D* was generated and given to the low-level assistive controller (Fig. 4). On the other hand, if we did not use this high-level controller, the desired velocity trajectory would not have been automatically modified to register the intention of the subject to pause the task. As a result, the velocity trajectory would follow the dashed line in Fig. 4. In such a case, when P1 wanted to start the task again, the desired velocity trajectory would start at point *C'* with non-zero velocity. This could create unsafe operating condition. In addition, since the desired velocity computation would not have included the pause action, restarting the task at point *C'* would not allow the completion of the task as desired. For example, in this case, if P1 had used the dashed velocity trajectory, she would start moving in the opposite direction at point *C'*. It could be possible to pre-program all types of desired velocity trajectories beforehand and retrieve them as needed. However, for non-trivial tasks such a mechanism might be too difficult to manage and extend as needed. The high-level controller provides a systematic mechanism to tackle such issues. It could also be seen that new velocity trajectories could be created dynamically using the generator block. In order to generate the required trajectories, the task parameters were needed. High-level controller monitored the progress of the task and made decision on the modification of the task parameters. When the subject reached the desired position, which is *0.3m*, and then velocity trajectory from *D* to *E* was generated and given to the low-level assistive controller so that P1 moved back to the starting position.

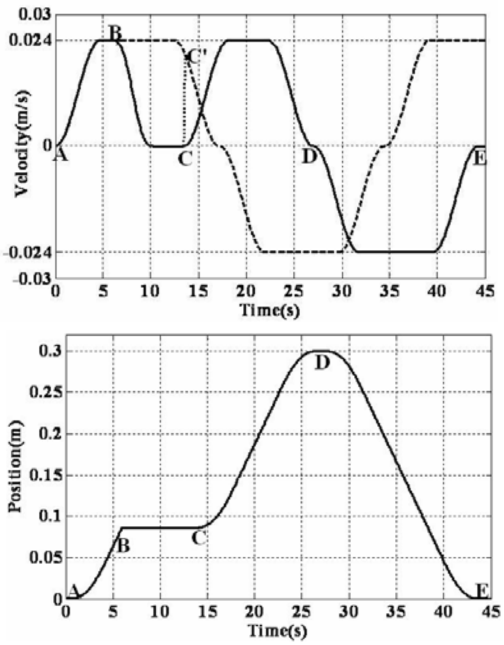


Figure 4: Motion Trajectories When Task is paused

As could be seen from the results from experiments 1 and 2, the high-level controller determines the task parameters dynamically based on patients' performance and monitors the safety related events to generate the necessary motion trajectories at the required time.

VI. Conclusion And Future Work

In this paper we present a new control approach to offer robotic assistance for stroke patients that include the coordination between a high-level controller and low-level assistive controller. The high-level controller can coordinate with a low-level assistive controller to improve the robotic assistance with the following objectives: 1) to monitor the upper arm rehabilitation task; and ii) to make necessary decisions to address the status of the task. We present a systematic design procedure for the high-level controller to accomplish the above objectives. Note that the proposed high-level controller can be integrated with other low-level controllers with minor modifications. We have conducted experiments with unimpaired subjects and demonstrated the usefulness of the high-level controller. The control architecture presented here is an example of a hybrid control system. There has been no work to our knowledge on designing such kind of control architecture for rehabilitation purposes.

An important direction for future development involves testing the usability of the proposed control architecture with stroke patients. New methods to detect human state information can be integrated into the control architecture such as ECG signals can be used to monitor patients' heart rate to detect their exhaustion and a voice recognition system can be integrated to examine the patient's verbal commands to modify the task.

ACKNOWLEDGMENTS

We gratefully acknowledge the help of Dr. Thomas E. Groomes who is the Medical Director of Spinal Cord and Traumatic Brain Injury Program, and therapist Sheila Davy of Vanderbilt University's Stallworth Rehabilitation Hospital for their feedback about task design during this work. The work was supported by Vanderbilt University Discovery grant.

REFERENCES

[1] American Heart Association: Heart and Stroke Statistical Update, http://www.Americanheart.org/statistics/stroke.htm, 2000.

[2] Matchar D.B., Duncan P.W.: Cost of stroke, Stroke Clin Updates, **5**, 9-12 (1994).

[3] Kahn L.E., Zygman M.L.Rymer W.Z, Reinkensmeyer D.J: Robot-assisted reaching exercise promotes arm movement recovery in chronic hemiparetic stroke: a randomized controlled pilot study. J. of NeuroEngineering and Rehabilitation. **3(12)**, 1-13 (2006).

[4] Lum, P.S.; Burgar, C.G.; Van der Loos, H.F.M.; Shor, P.C.; Majmundar M.; Yap R.2006. MIME robotic device for upper-limb neurorehabilitation in subacute stroke subjects: A follow-up study. J. of Rehab. Research & Development, **43(5)**, 631-642 (2006).

[5] Krebs H. I, Ferraro M., Buerger S.P, Newbery M. J., Makiyama A., Sandmann M., Lynch D., Volpe B. T., Hogan N.:Rehabilitation robotics: pilot trial of a spatial extension for MIT-Manus. Journal of NeuroEngineering and Rehabilitation. **1(5)**, 1-15, (2004).

[6] Loureiro R., Amirabdollahian F., Topping M., Driessen B., Harwin W.:Upper limb mediated stroke therapy - GENTLE/s approach. Autonomous Robots. **15**, 35-51 (2003).

[7] Koutsoukos X.D., Antsaklis P.J.:Hybrid Systems Control. ISIS Technical Report ISIS-2001-003 (2001).

[8] Antsaklis P.J., Koutsoukos X.D: Hybrid Systems: Review and Recent Progress. Chapter in Software-Enabled Control: Information Technologies for Dynamical Systems, T. Samad and G. Balas, Eds., IEEE Press. 1-29 (2003).

[9] D. Erol, N. Sarkar, "Design and Implementation of an Assistive Controller for Rehabilitation Robotic Systems," Inter. J. of Adv. Rob. Sys, **4** (2007).

[10] Antsaklis P.J.: A Brief Introduction to the Theory and Applications of Hybrid Systems. Proceedings of the IEEE on Special Issue on Hybrid Systems: Theory and Applications, **88(7)**, 879-887 (2000).

[11] Carey J., Durfee W., Bhatt E., Nagpal A., Weinstein S., Anderson K., Lewis S.: Tracking vs. movement Telerehabilitation training to change hand function and brain reorganization in stroke. Neurorehabilitation and Neural Repair (2006) (submitted).

[12] Stateflow, Mathworks Inc, http://www.mathworks.com/products/stateflow/?BB=1.