A Low Cost Telerehabilitation Device for Training of Wrist and Finger Functions After Stroke

ABSTRACT
There is a need for robotic rehabilitation devices that increase the outcome while reducing the cost of therapy. This paper presents a device for training of supination/pronation, dorsal wrist extension and finger manipulation after stroke. The system exhibits modularity in terms of the communication architecture and different optional components. User interfaces (UI) can be implemented on different kinds of devices including a Raspberry Pi single-board computer on which a Qt-based graphical UI was run. Telerehabilitation functionality is included using SSL-encrypted RESTful web services on a three-tier architecture. Expensive sensors were omitted in order to have a cost-effective system which is a requirement for home-based rehabilitation. The torque sensing is based on current measurements being evaluated by comparing it to force-torque sensor values. After canceling out the static friction, the low error justified the omission of an additional sensor.

Categories and Subject Descriptors
Applied computing [Life and medical sciences]: Consumer health

General Terms
Design, Measurement, Performance

1. INTRODUCTION
In high-income countries, stroke is the third most common cause of death and the major cause of acquired adult disability [14]. Besides personal consequences, this means a high economical impact. The average lifetime cost for stroke rehabilitation per case in Germany is 43,129 EUR and projected 3.4 million new cases of ischemic stroke from 2006 to 2025 lead to estimated costs of 108.6 billion EUR [5].

Often, the upper limb is affected with only 20% to 56% of all stroke survivors regaining useful function after three months [10]. In robotic therapy, training of the distal parts of the upper-limb of the wrist appears to be a crucial factor. A wrist extension for robotic upper-limb therapy device further increased the rehabilitation outcome in comparison to training of the shoulder and elbow alone [6]. Reaching and wrist supination/pronation training is particularly important since it promotes progress towards more functional whole upper extremity movements [11].

Several robotic systems for upper limb rehabilitation have been presented [8]. While some studies show promising results in outcome [13], they are not consistently in favor of robotics [7]. This underlines the necessity for a better understanding of motor learning which can be enhanced by means of robotic rehabilitation [12]. Seen from another perspective, these systems do not have to compete against traditional therapy but can extend it, for instance, by deploying rehabilitation devices in the home environment.

Home rehabilitation prospectively reduces costs by increasing independent training time and relieving the load of therapists. Telerehabilitation gives them a means of control and surveillance over the training and the possibility to intervene if necessary. However, most of the proposed devices are not suitable for home rehabilitation. Many systems are too expensive, e.g. due to the use of force-torque sensors [9], do not offer modularity and virtual rehabilitation [4], or are fitted to other systems [1]. 75% of the devices observed in a comprehensive review have not even undergone any sort of test due to high complexity and poor usability [2].

Based on the beforementioned points, we propose a device for training of wrist (supination/pronation/dorsal extension) and finger functions focusing on its application in the home environment. The system requires to be cost-effective, provide a user interface, actuator and sensors for virtual rehabilitation and the study of rehabilitation paradigms, particularly visual feedback distortion [3], and give remote access to session data for telerehabilitation.

2. CONCEPT AND IMPLEMENTATION
2.1 Concept and Design
There are two basic approaches of training devices: End-effectors and exoskeletons. Rotational movements involve a high number of the 27 degrees of freedom of the hand, which exoskeletons have to provide, in order to allow fully unconstrained movements. Following a cost-effective approach, we chose an end-effector design which allows tasks involving many DOF without the need for a high number of mecha-
2.2 Actuation, Sensing, and Electronics

A brushed motor with a nominal torque of 270 mNm, and a gear with a reduction gear ratio of 7:1 was used. Regarding the efficiency, the maximum total torque is 1.7 Nm. The combination of a strong motor with a low reduction gear ratio results in low backdrivability while achieving decent maximum torque. It is not sufficient to work against strong spasms, but safety concerns and backdrivability outweighed this possibility. An encoder with 1024 impulses per revolution delivers the relative angle in incremental steps of 0.05°. A VHN2SP30 H-bridge motor driver amplifies the pulse-width modulation (PWM) signals from the microcontroller and switches between the directions, breaking and coasting depending on the input pins.

Torque measurements are an important factor in assessing the patient’s capabilities. Force-torque sensors were not an option, since they cost many times more than the presented system. Therefore, the torque measurement utilizes the linear relationship to the armature current. A Hall effect based current sensor IC, dimensioned for the Ampere range of the motor, converts the current to a voltage measured by the microcontroller’s 10-bit analog-to-digital converter (ADC). The maximum torque can, then, be restricted by comparing the measured current to a set point and adjusting the PWM signal with an integrative controller. We use two different microcontroller boards that both use ATmega328P chips but differ in the UART transmission. One uses a CP2102 USB to UART bridge that allows a tethered USB connection to a virtual COM port on the PC. The second board uses a BTM-222 Bluetooth module with Serial Port Profile (SPP) to set up a serial connection.

2.3 Software and Communication

The software is designed in a modular way. It consists of a three-tier architecture to provide tele or home rehabilitation capabilities. The client, located at the patient’s side, gathers data for the evaluation of the rehabilitation progress. This information is preprocessed and transmitted to the server located at the clinic side.

To maximize the flexibility in client hardware, our prototype client software is a cross-platform implementation based on Qt4 which runs on standard PCs (Windows, Linux, MacOSX) as well as on single-board computers like the Raspberry Pi. The Qt4onPi group provides support for Qt4 and Qt5 on the Raspberry Pi. We ported and natively compiled our software on a Raspberry Pi Model-B running Raspbian wheezy to provide an all-in-one client. The software includes a GUI for progress measurements, motivating rehabilitation games and progress visualization. The interface can alternatively be controlled by the integrated navigational switch panel, touchscreen or standard PC input devices (keyboard/mouse).

The communication architecture allows for a variety of platforms to access the functionality and visualize sensor data. The client connects to the microcontroller alternatively over a wired or bluetooth serial UART connection to gather data or set parameters like torque or target position. The data of each training session is transmitted to the clinic side server or cached locally to be transmitted whenever a server connection is possible. With this implicit offline mode the training can be continued anywhere and anytime.
The server software is hosted on a Glassfish application server and implements a SSL-encrypted RESTful web service which receives data from training sessions and provides configuration values to the client software like targets. It consists of a web front-end based on the JavaServer Faces Framework PrimeFaces and connects to a JavaDB database. The RESTful approach combined with the PrimeFaces UI ensures the compatibility to various client and web-browsers. The interface features administrative, therapists’, and patient views with restricted rights.

3. PERFORMANCE EVALUATION

The first experiment evaluates the accuracy of the torque estimation from the current measurement. We use a force-torque sensor that is rigidly connected to the device. A current profile is applied on the end-effector and the estimated torque is compared to the measured one. The profile includes a step at the beginning and a drop at the end of the sampling period. These two points are used for synchronization and scaling of the two sensors that return samples with constant but different frequencies. The values after the initial step and before the increase of current define the baseline on which the force-torque sensor is calibrated. The results are plotted in Fig. 3.

The dotted line represents the initial setpoint current which is converted using the linear relation between torque and current. At the beginning of the slope, the increase of current does not result in higher torque. This is caused by unwanted influences of the gear, mostly by static friction. We determine the static friction together with the torque constant by fitting a linear model with $T(i) = k_m i + b$, where $T$ is torque, $i$ is current and $k_m$ is the torque constant, to the torque measurement. Iteratively, we increase the static friction and repeat the fitting without the values below the threshold until $b$ is approximately zero. The static friction was determined to be 188 mA. The slope represents the torque constant. With 50.77 mNm/A, it lies close to the quotient of the starting current and the stall torque of 52.75 mNm/A from the datasheet.

Then, we determined the static friction with the traditional method and compared it to our former result. The current was increased until a movement occurred, determined by an encoder value unequal to zero. After repeating 100 runs, the measured threshold currents were averaged and resulted in a breakaway current of 120 mA. The lower value comes from the encoder’s high sensitivity in conjunction with the gear’s backlash. Since the application of the higher threshold current onto the motor does not lead to unwanted continuous movement, the higher value is valid. It can now be utilized to decrease the initial user induced torque and improve the accuracy of the torque estimation. We measured a root-mean-square error of 25.78 mNm for the fitted dataset. Applied to the measurements from a second run, the error was only slightly higher with 30.08 mNm.

The torque control can be applied in different training modes. We implemented a virtual spring where the torque or the current set point, respectively, is linearly increased with the angle difference to a neutral position. A representative short session is shown in Fig. 4 with the angle, velocity, and torque plotted against the samples of 10ms length.

4. CONCLUSION

A rehabilitation device for training wrist and finger functions has been proposed. The torque estimation based on current measurements and the angle sensing from the motor encoder readings permits virtual rehabilitation and studying rehabilitation paradigms. The performance evaluation revealed a relative root-mean-square error of 2.2% which is acceptable considering the saved cost by omitting a force-torque sensor. It means that expensive sensors can be omit-
Figure 4: Representative short session in virtual spring mode.

5. REFERENCES