2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Cooperative Control of a Compliant Manipulator for Robotic-Assisted Physiotherapy

Gauthier Gras, Valentina Vitiello, *Member, IEEE* and Guang-Zhong Yang, *Fellow, IEEE*

Abstract— **In recent years, robotic systems have been playing an increasingly important role in physiotherapy. The aim of these platforms is to aid the recovery process from strokes or muscular damage by assisting patients to perform a number of controlled tasks, thus effectively complementing the role of the physiotherapist. In this paper, we present a novel learning from demonstration framework for cooperative control in robotic-assisted physiotherapy. Unlike other approaches, the aim of the proposed system is to guide the patients to optimally execute a task based on previously learned demonstrations. This allows the generation of patient-specific gestures under the supervision of the expert physiotherapist. The guidance is performed through stiffness control of a compliant manipulator, where the stiffness profile of the generalized trajectory is determined according to the relative importance of each section of the task. In contrast with the traditional learning approach, where the execution of the generalized trajectory by the robot is automated, this cooperative control architecture allows the patients to perform the task at their own pace, while ensuring the movements are executed correctly. Increased performance of the learning framework is accomplished through a novel fast, low-cost multi-demonstration dynamic time warping algorithm used to build the model. Experimental validation of the framework is carried out using an interactive setup designed to provide further guidance through additional visual and sensory feedback based on the task model. The results demonstrate the potential of the proposed framework, showing a significant improvement in the performance of guided tasks compared to unguided ones.**

I. Introduction

Every year, an estimated 15 million people suffer from strokes worldwide, 5 million of which result in death [1]. For stroke survivors, the symptoms can range from a few observable signs to complete paralysis [2]. Post-stroke rehabilitation is important to the functional recovery and independence of the patient. National clinical guidelines for stroke consider physiotherapy highly beneficial, and it is now widely accepted as a fundamental part of the process of stroke recovery [3]. Stroke cases aside, physiotherapy also plays a central role in aiding rehabilitation from multiple types of common injuries. It is frequently used in cases of muscular damage or impairment, in particular sport-related injuries such as tendinitis of the long head of the biceps [4]. Therefore, further improvements of the methods used in physiotherapy and development of dedicated technologies could lead to considerable health and economic benefits.

With the maturity of robotics technology, current rehabilitation platforms are increasingly relying on robot assisted rehabilitation. Systems such as the Lokomat$^{\text{TM}}$ [5], the WalkTrainer$^{\text{TM}}$ [6], or the MIT-Manus$^{\text{TM}}$ [7] are examples of robotic platforms designed to assist patients during rehabilitation. The general principle behind these systems is to aid the patients to perform a task correctly by guiding them during its execution. While this allows for significant improvements in recovery time, the accurate modeling and generation of these gestures remains challenging.

The aim of the work presented in this paper is to create a robotic framework for physiotherapy that is both easy to use by therapists and provides accurate guidance to the patients. The use of learning from demonstration (LfD) techniques allows tasks to be demonstrated directly to the robot by the patient under the supervision of the therapist. This provides the therapist with a hands-on means of generating new and tailor-made exercises. Once the demonstrations are recorded, the system aligns the demonstrations in time and uses this information to build a model of the task. A regression of the model can then be performed, generating an optimal execution trajectory for the patient. In practice, this framework implements a direct skill transfer from the expert therapist to the compliant robot, so that the patient can then continue the exercise without the expert supervision. In this manner, the patient can perform tasks as if directly guided by the therapist, but with the additional advantages of a robotic platform.

The proposed framework is described in details in the following sections. Firstly, Section II introduces the methods used to model the task based on the recorded demonstrations and to generate the corresponding stiffness profile for the cooperative controller. The novel fast multi-demonstration hierarchical dynamic time warping algorithm used to align the demonstrations in time is also presented. Section III gives an overview of the complete robotic system, highlighting the different components used and the underlying control architecture. The experimental protocol designed for the validation of the framework is presented in Section IV, while the experimental results are discussed in Section V. Finally, Section VI concludes the paper by highlighting the advantages of the proposed framework for robotic-assisted physiotherapy.

II. Methodology

In order for a task to be repeated by a patient under the guidance of the compliant robot, a cooperative control policy must be established. This is achieved through six main steps using the proposed LfD framework, as presented in Fig. 1. After recording, the demonstrations must be aligned in

Gauthier Gras, Valentina Vitiello, and Guang-Zhong Yang are with the Hamlyn Centre for Robotic Surgery, Imperial College London, SW7 2AZ, London, UK (email: gauthier.gras12@imperial.ac.uk).

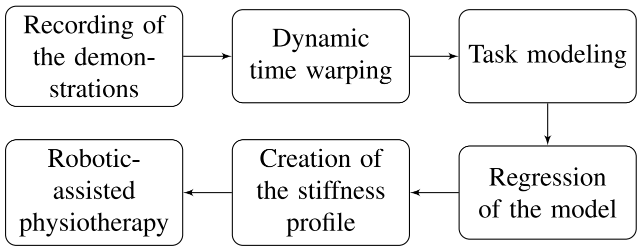

Recording of the demonstrations
Dynamic time warping
Task modeling
Regression of the model
Creation of the stiffness profile
Robotic-assisted physiotherapy

Fig. 1. Architecture of the proposed learning from demonstration framework for robotic-assisted physiotherapy.

time to generate meaningful data for the modeling of the task, which is obtained using a Gaussian Mixture Model. Subsequently, a time regression is performed resulting in a generalized trajectory. Lastly, a stiffness profile is associated with the generalized trajectory, to provide guidance during the task execution.

A. Dynamic Time Warping

As mentioned above, the recorded demonstrations need to be aligned in time to extrapolate the main features of the task correctly. To this end, a novel hierarchical dynamic time warping (DTW) algorithm is introduced, as described in the following.

1) Standard approach: In its standard formulation, the DTW of two time dependent signals $\boldsymbol{X} : \{x_1, ..., x_N\}$ and $\boldsymbol{Y} : \{y_1, ..., y_M\}, (M, N) \in \mathbb{N}^2$ consists in finding the optimal time alignment between their elements. To this end, a cost function c is used to compare the different features of these signals which is defined for any two points of $\boldsymbol{X}$ and $\boldsymbol{Y}$ as a measure of distance:

$$c(x_i, y_j) = |x_i - y_j|, \ (i, j) \in [1 : N] \times [1 : M]. \quad (1)$$

A cost matrix C can therefore be defined, whose entries correspond to the above defined cost values:

$$C(i, j) = c(x_i, y_j), C \in \mathbb{R}^{\mathbb{N} \times \mathbb{M}}. \quad (2)$$

Each element x_i of $\boldsymbol{X}$ can then be matched with a single element y_j of $\boldsymbol{Y}$ by tracing a path through the cost matrix C. Such alignment path w of length L is thus defined as a series of indexes from C:

$$w = \{(n_1, m_1), (n_2, m_2), \ldots, (n_L, m_L)\}, \quad (3)$$
$$(n_l, m_l) \in [1 : N] \times [1 : M] \ \forall l \leq L.$$

The total cost d of an alignment path w of length L can therefore be easily calculated as the sum of the cost values associated with each index within the path:

$$d_w = \sum_{l=1}^{L} c(x_{n_l}, y_{m_l}). \quad (4)$$

Finally, the optimal path $w_{optimal}$ is found as the least total cost path that satisfies the above restrictions. The search for the optimal path is conducted by building an Accumulated Cost Matrix (ACM) D. This matrix is an $N \times M$ matrix similar to the cost matrix C. However, each element of the ACM adds the minimum value of the surrounding previous elements to the cost value between two points of $\boldsymbol{X}$ and $\boldsymbol{Y}$:

$$D(i, j) = C(i, j) + \begin{cases} B \text{ if } i > 1, j > 1, \\ D(i, j-1) \text{ if } i = 1, j > 1, \\ D(i-1, j) \text{ if } i > 1, j = 1, \\ 0 \text{ otherwise.} \end{cases} \quad (5)$$
$$B = min(D(i-1, j), \ D(i, j-1), \ D(i-1, j-1)).$$

After computation of the ACM, the optimal path $w_{optimal}$ can be deduced. Since $w(L) = (N, M)$, the optimal path can be traced back entry by entry starting from the last element of the matrix, (N, M). The optimal path is entirely known when the first element $D(1, 1)$ is reached ($l = 1$). For a detailed review of the standard DTW algorithm, please refer to [8] and [9].

2) Generalization: The DTW problem of the alignment of two time-dependent signals can be generalized to consider more than two multi-dimensional signals. Consider N K-dimensional signals $\boldsymbol{X_1}, \boldsymbol{X_2}, ...\boldsymbol{X_N}$, with $N \in \mathbb{N}$ and defined as follows for $(i, j) \in \mathbb{N}^2, i \leq N, j \leq K$:

$$\boldsymbol{X_i} = \{\boldsymbol{Y_{i_1}}, \boldsymbol{Y_{i_2}}, ..., \boldsymbol{Y_{i_K}}\}, \quad (6)$$

$$\boldsymbol{Y_{i_j}} = \{y_{i_{j_1}}, y_{i_{j_2}}, ..., y_{i_{j_{s_i}}}\}, (y_{i_{j_1}}, y_{i_{j_2}}, ..., y_{i_{j_{s_i}}}) \in \mathbb{R}^{S_i}, \quad (7)$$

where $S_i \in \mathbb{N}$ is the size of the dimensions within signal number i. The signals $\boldsymbol{X_i}$ to be aligned will be referred to as demonstrations, composed of dimensions $\boldsymbol{Y}$. Since the demonstrations of one task are collected in the same manner, they must all have the same number of dimensions, and the size (number of elements S_i) should be constant for each dimension of a given demonstration. Consequently, in this case performing the DTW consists of finding the optimal alignment of all the dimensions $\boldsymbol{Y}$ within each of the demonstrations $\boldsymbol{X}$. The cost function must therefore be modified to take into account the different dimensions and demonstrations.

- *Generalization to multiple dimensions only:* The cost function defined by (1) is a measure of distance between two 1-dimensional points (a simple difference). Generalized to K dimensions, the cost for two time-points s_1 and s_2 of $\boldsymbol{X_1} = \{\boldsymbol{Y_{1_1}}, \boldsymbol{Y_{1_2}}, ..., \boldsymbol{Y_{1_K}}\}$ and $\boldsymbol{X_2} = \{\boldsymbol{Y_{2_1}}, \boldsymbol{Y_{2_2}}, ..., \boldsymbol{Y_{2_K}}\}$ becomes a measure of distance between two K-dimensional vectors:

$$c(s_1, s_2) = \sqrt{\sum_{k=1}^{K} (y_{1_k}(s_1) - y_{2_k}(s_2))^2}, \quad (8)$$
$$(s_1, s_2) \in [1 : S_1] \times [1 : S_2],$$

- *Generalization to multiple demonstrations:* The measure of distance between two K-dimensional vectors can be extended to a measure of distance between N K-dimensional vectors by considering the sum of the distances between all demonstrations taken two by two. Thus the total generalized cost function for

N demonstrations of K dimensions at the time-points $(s_1, ..., s_N) \in \mathbb{R}^{S_1 \times S_2 \times ... \times S_N}$ is given as:

$$c(s_1, ..., s_N) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} ||\boldsymbol{X_i}(s_i) - \boldsymbol{X_j}(s_j)||. \quad (9)$$

As the ACM is built directly from the cost measure, it can also be generalized to multiple dimensions and demonstrations. Each additional demonstration also adds a dimension to the ACM. The ACM can be constructed in a similar way to the case of two 1-dimensional signals, however the choice of the minimum surrounding past value must consider all dimensions. Let $D(s_1, ..., s_N)$ be the element of the ACM for which the minimum adjacent past value element must be found. The boundary problem is initially not considered: $\forall i \in [1 : N]$, $s_i > 1$. Finding all the adjacent past values of $D(s_1, ..., s_N)$ is similar to finding all the possible permutations of an N bit binary number. This is done in the presented algorithm by considering $D(s_1, ..., s_N)$ to be the *all ones* state of the corresponding number. There are thus a total of 2^{N-1} possible cases. The permutations are parsed with a bitwise counter: the starting combination is *all zeros* and all counters also start at zero. The n^{th} bit is permuted when its counter reaches 2^n, at which point its counter is reset to zero. This method is illustrated for 3 demonstrations in Table I.

To take into account the boundary problem, the number of bits that cannot be permuted (*i.e.* when the corresponding value $s_i = 1$) must be tracked. During the permutation process, each bit must be checked to see if it is permutable, and otherwise it should be skipped. The total number of skipped permutations for u unpermutable bits is equal to $2^{N-u-1} \times (2^u - 1)$. This should be taken into consideration when performing the permutations as it can save a substantial amount of computing time, particularly when windowing techniques are used. In this manner, the fully generalized ACM can be computed.

3) Hierarchical approach: While the standard DTW algorithm can be generalized to multiple dimensions and demonstrations as described above, the exponential increase in computing time and usage of resources make this approach difficult to implement in a real scenario. Windowing techniques such as the Sakoe-Chiba band and the Itakura parallelogram [9] can be used to reduce the necessary computational time. However, these algorithms are prone to errors for signals with large delays, as the optimal path will be offset from the diagonal of the ACM. In this paper, a hierarchical approach inspired from [10] is introduced, which can be generalized in an efficient manner to multiple demonstrations.

TABLE I
MINIMUM ADJACENT PAST VALUE PARSING FOR A 3 DIMENSIONAL ACM ELEMENT $D(s_1, s_2, s_3)$

Binary representation	ACM element
000	$D(s_1-1, s_2-1, s_3-1)$
001	$D(s_1-1, s_2-1, s_3)$
010	$D(s_1-1, s_2, s_3-1)$
011	$D(s_1-1, s_2, s_3)$
100	$D(s_1, s_2-1, s_3-1)$
101	$D(s_1, s_2-1, s_3)$
110	$D(s_1, s_2, s_3-1)$
111	$D(s_1, s_2, s_3)$

Hierarchical DTW is a windowing technique that strives to have the window centered on the optimal path, as opposed to the diagonal of the cost matrix. Since standard DTW of multiple large signals is computationally expensive, the aim of this technique is to obtain an estimate of the optimal path from undersampled versions of the signals. A first estimate of the optimal path is generated by applying the standard DTW algorithm on a highly undersampled version of the signals (*i.e.* by a factor of 16). This path is then projected onto a less undersampled space of the signals (*i.e.* by a factor of 8). A window is applied around the projected path, and a new optimal path is obtained from the computation of the ACM within that window (all other entries are assigned a cost of infinity). The process is then repeated until the final optimal path is obtained. The algorithm is illustrated for two signals $\boldsymbol{X_1}$ and $\boldsymbol{X_2}$ in Fig. 2(a)-(e). The corresponding result obtained using the generalized version of the standard DTW is shown for comparison, in Fig. 2(f). In the presented algorithm the undersampling factors are chosen according to a geometric series of constant reason referred to as the undersampling coefficient, typically equal to 2.

In order to avoid parsing the entire ACM, and thus reduce the required computational time, the N-dimensional cost matrix D is represented using linear indexing as a long 1-dimensional array D_{linear} of size D_{length}, only reporting the entries within the applied window. This also reduces the resource requirements of the algorithm since only a small number of elements are stored in memory. Since the window is known, it is possible to find two mapping functions $indexTosubscripts$ and $subscriptToindex$ such that, given an element of D as $(i_1, ..., i_N)$ and l as the corresponding linear index of D_{linear}:

$$(i_1, ..., i_N) = indexTosubscripts(l) \quad (10)$$

$$l = subscriptToindex((i_1, ..., i_N)) \quad (11)$$

The expression of these functions depends on the manner in which the window is generated and is not detailed here. In conclusion, the main advantages of this approach are that:

- The window is centered on the optimal path at each iteration.
- The computational cost is reduced drastically because of the undersampling and windowing.
- The resource allocation is minimized through the use of a linear indexing.

For the proposed DTW algorithm, Fig. 3 presents a comparison of the hierarchical approach against a generalization of the standard approach, highlighting how the improved

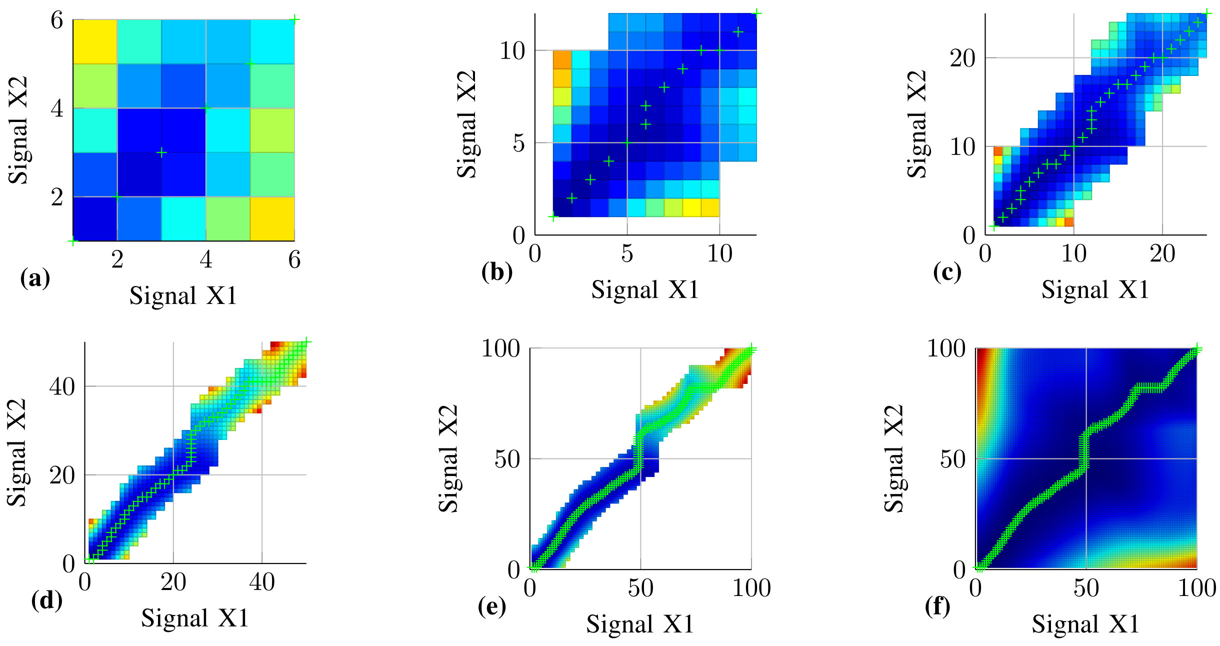


Fig. 2. Hierarchical dynamic time warping for two signals: (a) Undersampling factor: 16, (b) Undersampling factor: 8, (c) Undersampling factor: 4, (d) Undersampling factor: 2, (e) No undersampling, (f) Standard DTW algorithm. Green crosses: points of the optimal path. Low ACM value: dark blue. High ACM value: red. ACM value not computed: white.

performance of the proposed algorithm becomes more and more significant as the number of points in the demonstrations increases. This algorithm was consistently used in the experiments described hereafter, and proved robust with computing times of under twenty seconds (typically with 3 demonstrations of approximately 500 points). Lastly, the optimal path obtained with the hierarchical DTW was identical to the one obtained with the standard algorithm when the two were compared in a series of tests with experimental trajectories, proving that the applied window is always centered on the optimal path.

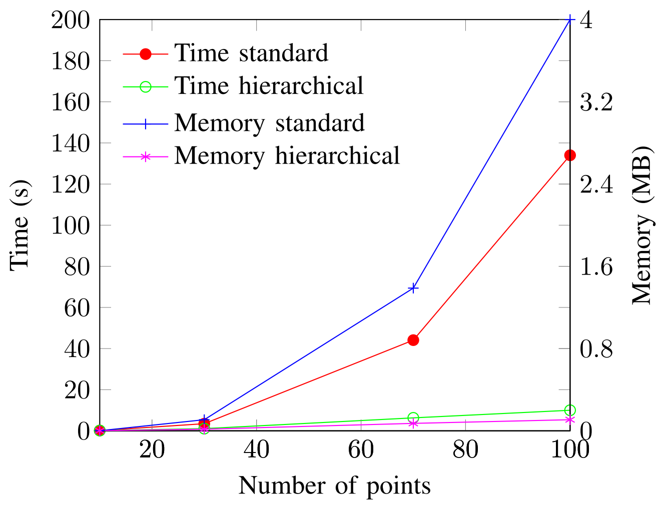


Fig. 3. Time and memory usage comparison: hierarchical approach against standard approach for three demonstrations of n points each.

B. Task encoding

Once the recorded demonstrations are aligned in time, a generalized model of the task can be extracted using a standard LfD technique. In this paper, a Gaussian Mixture Model (GMM) was chosen to model the task trajectories. As the aim of the proposed framework is to guide the patient through the generalized trajectory rather than automating the execution of the task, a stiffness controller is then implemented based on the covariance information provided by the regression of the task model.

1) Gaussian Mixture Modeling: As GMMs are commonly used for task encoding and the implementation used in this paper follows the standard Expectation-Maximization algorithm presented in [11] and [12], their formulation will not be discussed in detail. Two criteria are usually considered to determine the optimal number of states in the model: the Bayesian Information Criterion (BIC) and the Akaike Information Criterion (AIC), as expressed in equations (12) and (13) respectively.

$$BIC = -2 \times ln(Lh) + k \times ln(n), \quad (12)$$

$$AIC = -2 \times ln(Lh) + 2 \times k, \quad (13)$$

- n: number of observations in the model.
- Lh: likelihood of the estimated model.
- k: number of free parameters (states) to be estimated.

When the above criteria do not converge towards the same number of states, the former is designed to favor low dimensionality solutions, while the latter would result in higher dimensionality ones. For the specific application presented in this paper, overfitting models are considered more acceptable than underfitting ones. This is because

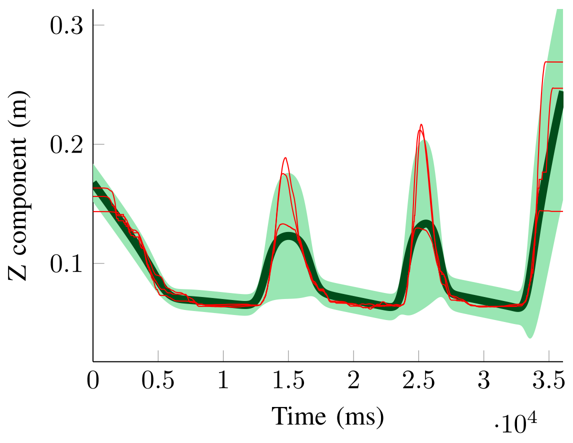


Fig. 4. Covariance information from the time-regression of a GMM. Red: demonstration data after dynamic time warping. Dark green: time-regression of the model. Light green: covariance indicator.

the movements performed in physiotherapy do not require millimeter precision, so slight over-smoothing of the task trajectory is acceptable. On the other hand, jerky movements could potentially be dangerous for the patient, depending on the level of injury.

2) Stiffness control: The data obtained from the time-regression of a GMM is referred to as the generalized trajectory. This optimal path can be used to guide a patient during the execution of the task through cooperative control of a robotic system. Based on the robot end-effector's position, it is possible to track the current point of the task trajectory executed by the patient, and update it at every cycle of the main control loop. This position can then be mapped to a corresponding point of the generalized trajectory, typically the closest one to the end-effector's current position. In order to guide the user towards the generalized trajectory, the robot has to possess a non-zero stiffness value. At the same time, the patients should be free to perform the task at their own pace, therefore the movement should not be automated. The idea is to use stiffness control as described in [13] to ensure the patient is performing the task correctly. Instead of following a trajectory automatically, however, the robot is set in a semi-compliant mode. By default the robot is in a gravity compensation mode and fully compliant. When moving away from the generalized trajectory, the robot gradually stiffens. This variation in stiffness is generated by a spring-damper system controller with stiffness coefficient k. The higher the value of k, the higher the stiffness, and thus the more force the robot will exert to bring the end-effector towards the closest position on the generalized trajectory.

In the presented framework, this stiffness coefficient is defined for each point of the generalized trajectory. The desired stiffness profile is therefore created by assigning a value to k_i, $\forall i \in [1 : S]$, with S the size of the generalized trajectory. The values of the stiffness are generated according to the covariance information from the demonstrations. The following logic is used:

- If the demonstrations are close together in a section of the task (after DTW), then that section is deemed of critical importance. The stiffness is set high to promote correct movements.
- If the demonstrations are spread apart in a section of the task (after DTW), then that section is deemed of low importance. The stiffness is set low to promote freedom of action.

The use of the covariance information to generate the stiffness profile is illustrated in Fig. 4. In this case, the task consisted in drawing the letter 'H' three times using the robot end-effector. The three horizontal bars in the model correspond to the three bars of the H, while the parts in between correspond to the lifting of the end-effector from the paper to move to the next bar. Since the position of the end-effector during these movements is not always the same, the demonstrations are naturally spread apart, resulting in the robot becoming more compliant when these regions are reached during cooperative control. It should also be noted that the stiffness profile is axis-specific, therefore the robot can be compliant along one Cartesian axis and stiff along a different axis (the same applies for rotations).

III. System Overview

The presented framework comprises four main components, displayed in Fig. 5:

- The KUKA lightweight robot by KUKA Roboter GmbH (Augsburg, Germany).
- A Polaris Vicra optical tracking system by NDI (Waterloo, Ontario, Canada).
- An Arduino Nano micro-controller by Arduino (manufactured by Gravitech, CA, USA), linked to a vibration device.
- A control and computing system used to implement the LfD framework, generate the graphical interface, and handle the proper communication and interactions between the various system components.

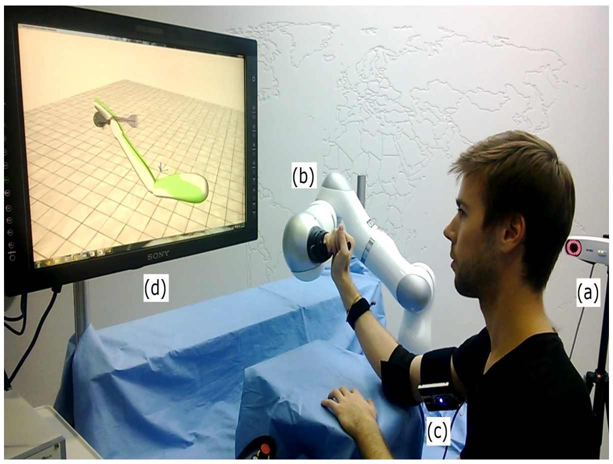


Fig. 5. Experimental system setup: (a) Polaris Vicra optical tracker, (b) KUKA lightweight robot, (c) Polaris optical markers and Arduino Nano board with the vibration device attached to armbands, (d) Graphical interface connected to the control system.

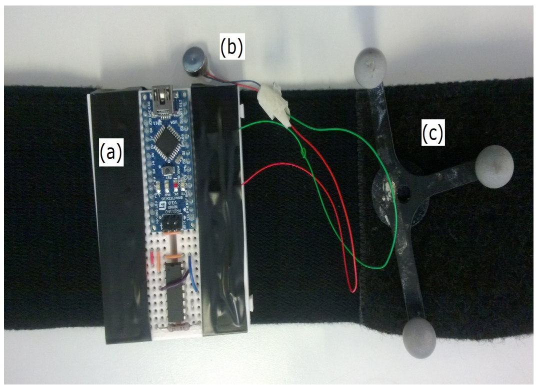


Fig. 6. Arduino Nano micro-controller, vibrator and Polaris marker attached to an armband. (a) Arduino Nano micro-controller on a breadboard, (b) Vibrator, (c) Polaris marker.

A. *KUKA Lightweight Robot*

The KUKA lightweight robot (LWR) is a 7 degrees-of-freedom robotic arm, equipped with a custom-made handle at the end-effector. The robot is programmed using a Cartesian impedance position controller, acting like a spring damper system centered on the commanded position.

B. *Polaris Vicra Optical Tracking System*

The Polaris Vicra optical tracking system uses infra-red technology to track the position of optical markers within its camera field of view. In the presented framework, three Polaris markers are used to determine the position of the user's arm in 3D space.

C. *Arduino Nano Micro-controller and Vibration Device*

The Arduino Nano is a micro-controller board equipped with a mini-USB port. In the presented setup this is mounted on a breadboard and linked to the control system via a mini-USB/USB cable. A vibrator is connected between the digital pin D9 of the board and one of its ground pins. The output of the pin D9 is controlled remotely by the control system, and can vary between 0 and 5 Volts. The vibrator is still at 0 Volts, and vibrates at maximum intensity at 5 Volts. Although not tested, the vibration increases in a roughly linear fashion. A picture of the Arduino Nano micro-controller, the vibrator and an optical marker attached to an armband is shown in Fig. 6.

D. *Control System*

The control framework is implemented in a C++ program running on a 2.3 GHz intel i5-430M CPU and GeForce GTS 360M GPU. The operating system used is Windows 7, although the entirety of the code is cross platform to be used with a Linux-based operating system. The C++ program handles all the main routines of the setup, including the recording of the demonstrations, DTW, learning algorithms, cooperative control modes, communication threads between different components, registration between the Polaris frame and the KUKA frame, and the graphical user interface.

E. *Experimental Setup*

For the experiments, the three Polaris markers and the Arduino Nano are fixed on the external side of three armbands attached to the arm of the subject. The vibrator is placed between one of the armbands and the arm of the subject. The final setup is shown in Fig. 7. The Polaris is integrated in the system in order to track the movements of the arm during task execution and provide further guidance to the user. The 3D position of the arm is determined on the basis of the current position of the Polaris markers and included into the task model. As a result, the generalized trajectory obtained from the time regression also contains the corresponding 3D position of each of the markers. This information is used to provide the user with visual feedback about the correctness of the arm position during task execution, as illustrated in Fig. 8. The optimal position of the user's arm at a specific point along the task is displayed in transparent green. The current position of the user's arm is shown in solid white. When the user moves, both models are updated. If the user moves too far from the optimal position, the corresponding sections of the solid white arm turn red.

To further emphasize the importance of the position of the arm during task execution, the vibrator is used to give sensory feedback to the user when he/she moves away from the correct position. The further the user moves away from the correct arm position, the higher voltage is applied to the vibrator, resulting in a more intense vibration.

IV. Experimental protocol

Three tasks commonly performed during upper-body physiotherapy are used to validate the proposed framework:

- Task 1 is a shoulder forward flexion.
- Task 2 is a shoulder abduction.
- Task 3 is a biceps flexion/relaxation.

Each participant demonstrates each task three times. During the demonstrations the robot is in a fully compliant gravity compensation mode. The participants are given instructions, and shown how to do the task accurately. Once three demonstrations of the task are recorded, the model learning is initiated, creating a generalized trajectory for

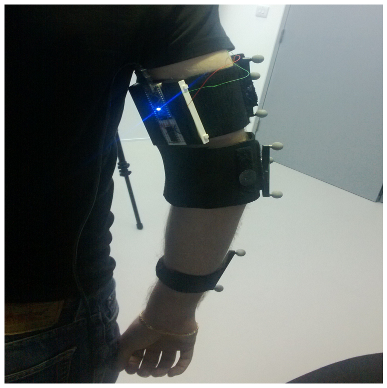
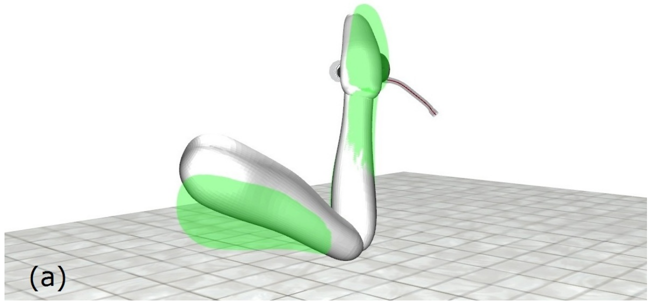


Fig. 7. Three Polaris optical markers and the Arduino Nano micro-controller mounted on armbands.



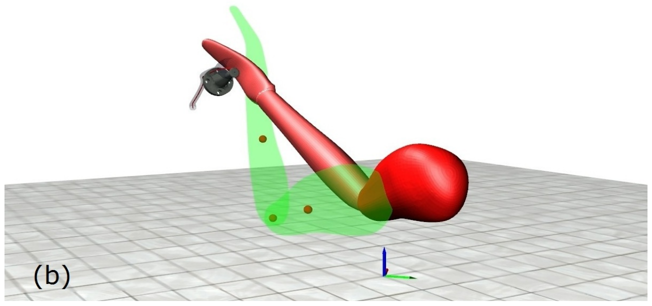


Fig. 8. Graphical display of the current position of the user's arm (in solid), and the optimal position of the user's arm for that point of the trajectory (in transparent green). The red spheres represent the positions of the Polaris markers at the time of the demonstration recording. (a) Good position, (b) Bad position, the arm turns red to notify the user of the incorrect posture.

the KUKA robot end-effector as well as for the optical markers. After recording a task, the subjects move on to the second phase of the experiment. During this phase the KUKA robot simulates a weight, similar to those used in physiotherapy. The mass of the weight is changed from subject to subject, but always set so that the subject requires a real effort to move the robot. This approach is justified with two arguments:

- Weights are used in physiotherapy for muscular rehabilitation and strengthening. The average weight used in the experiments was 3 kg.
- Since the participants were all healthy, the additional weight helped to simulate a physical impairment.

The subjects are then asked to repeat each task under two different control modes, once with the guidance provided by the framework, once without. The data from both these iterations is recorded. Note that, while during the demonstrations the task had to be done precisely once (*i.e.* precisely one 90° shoulder flexion), during these tests the subjects perform the movement several times to increase the size of the data collected. The protocol for each subject is summarized in Fig. 9.

V. Results

A total of 8 subjects took part in the experiment, resulting in 72 recorded demonstrations. The performance of the guided and unguided task execution for each participant is assessed in comparison to the corresponding optimal generalized trajectory. At each point of the trajectory, four error

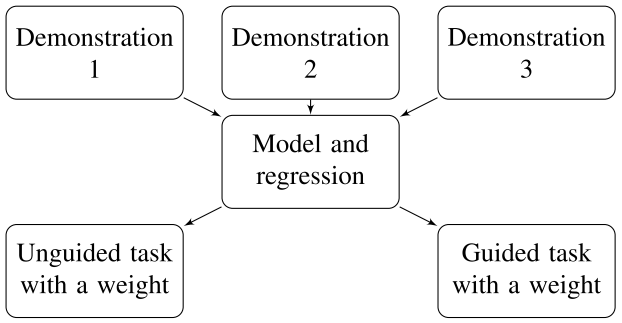


Fig. 9. Experimental protocol for the validation of the proposed learning framework.

measurements are calculated for the guided and unguided tasks: $\Delta_K, \Delta_{T1}, \Delta_{T2}, \Delta_{T3}$. These values respectively represent the error at that point between:

- the current end-effector position and the optimal end-effector position,
- the current wrist position and the optimal wrist position (marker 1),
- the current elbow position and the optimal elbow position (marker 2),
- the current upper arm position and the optimal upper arm position (marker 3).

The error data for the guided and unguided tasks is calculated by combining the recordings of all the participants. Thus for each task, four sets of data are available to compare the guided and unguided performance. The p-value of each dataset is calculated for each task with a ranksum test. The medians of $\Delta_K, \Delta_{T1}, \Delta_{T2}, \Delta_{T3}$ for each task are also calculated and reported in Table II. Finally, a boxplot of the 4 types of error data over all the tasks is shown in Fig. 10.

TABLE II
MEDIANS OF THE ERROR MEASUREMENTS FOR THE GUIDED AND UNGUIDED TASKS. P VALUE FOR ALL TASKS IS HIGHLY SIGNIFICANT

Shoulder flexion	Guided (mm)	Unguided (mm)
End-effector	4	30
Wrist marker	24	93
Elbow marker	31	66
Upper arm marker	32	52
Shoulder abduction	Guided (mm)	Unguided (mm)
End-effector	4	53
Wrist marker	22	103
Elbow marker	26	77
Upper arm marker	27	64
Biceps flexion	Guided (mm)	Unguided (mm)
End-effector	4	33
Wrist marker	14	48
Elbow marker	17	22
Upper arm marker	21	24

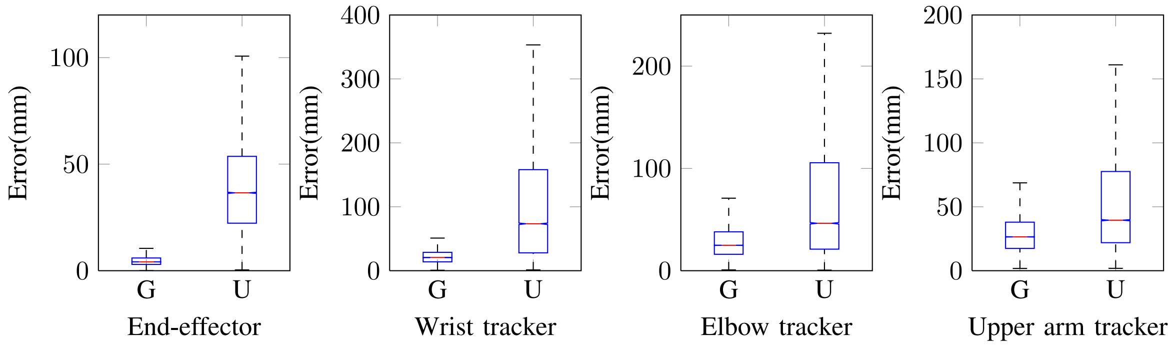


Fig. 10. Analysis of the error parameters for guided (G) and unguided (U) task executions using the data collected with 8 participants.

Due to the stiffness control, the end-effector only presents a sub-centimeter positioning error during guidance. The relative error is overall smaller for the upper arm marker as it is placed very close to the shoulder level and thus only executes limited motion. This is particularly true in task 3: biceps flexion and relaxation, where the elbow and the upper arm are supposed to remain static. These preliminary results are on the whole very positive, since the guidance generates a significant improvement of task performance reflected by the decreased error values. This effectively demonstrates the potential of the proposed learning framework, especially considering that all the participants in the study were healthy, and therefore had no muscular impairment preventing them from naturally executing the movements correctly. The following can be noted from the observed experimental results:

- The vibration feedback increases alertness when moving too far from the generalized trajectory. While a complete study was not conducted at that time, an opinion poll of the participants who tested the system before and after the integration of vibration feedback revealed that this addition highly improved awareness to staying close to the model.
- The weight used in the test is sufficient to cause a deviation from the optimal generalized trajectory without guidance. The value of the weight was set so that the participants felt they were conducting the same type of workout of a real exercising session.

VI. Conclusion

In this paper, we have presented a cooperative control framework for robot assisted physiotherapy. The principal advantage of the proposed system is the novel use of LfD techniques for robotic-assisted physiotherapy. This approach to task modeling enables the therapist to generate multiple different exercises with a single robotic system. No programming skills are required to create a new task, and only a minimal effort is required to configure the desired input and output parameters of the model. The use of a compliant manipulator is particularly suitable for the learning approach, since this allows intuitive and hands-on patient-specific task teaching, while a variety of tools can be mounted on the robot end-effector to perform different tasks. The combination of these advantages makes the platform presented in this paper a versatile and powerful tool for robotic-assisted physiotherapy.

The potential of the proposed framework is further demonstrated by the results obtained from preliminary experiments carried out over three tasks used in upper-limb physiotherapy. In all three tasks, the guidance provided by the platform through visual and tactile feedback generated a significant performance improvement. The results are particularly promising considering that all participants were healthy. It is therefore reasonable to expect even more pronounced improvements with a population of injured participants, where the guidance is actually required to ensure faster rehabilitation.

References

[1] World Health Organization, "World health statistics 2012," Tech. Rep. 2, 2012.

[2] R. D. Z. Joel Stein, Richard L. Harvey, Richard F. Macko, Carolee J. Winstein, *Stroke recovery and rehabilitation*. Demos Medical Publishing, 2008.

[3] N. S. Foundation, "Clinical Guidelines for Stroke and TIA Management," *National Stroke Foundation*, pp. 1–8, 2010.

[4] Hans K. Uhthoff, "Shoulder Injury and Disability," *The Workplace Safety and Insurance Appeals Tribunal, University of Ontario*, 2010.

[5] S. Jezernik, G. Colombo, T. Keller, H. Frueh, and M. Morari, "Robotic orthosis lokomat: a rehabilitation and research tool," *Neuromodulation : journal of the International Neuromodulation Society*, vol. 6, pp. 108–15, Apr. 2003.

[6] M. Bouri and Y. Stauffer, "The WalkTrainer: a robotic system for walking rehabilitation," *International Conference on Robotics and Biomimetics, Kunming, China*, pp. 1616–1621, 2006.

[7] N. Hogan and H. Krebs, "MIT-MANUS: a workstation for manual therapy and training. I," *International Workshop on Robot and Human Communication*, pp. 161–165, 1992.

[8] M. Müller, "Dynamic Time Warping (DTW)," in *Information Retrieval for Music and Motion*, ch. 4, pp. 231–231, 2007.

[9] P. Senin, "Dynamic time warping algorithm review," *Honolulu, USA*, pp. 1–23, 2008.

[10] S. Salvador and P. Chan, "FastDTW : Toward Accurate Dynamic Time Warping in Linear Time and Space," *Intelligent Data Analysis*, vol. 11, no. 5, pp. 561–580, 2007.

[11] S. Calinon, F. Guenter, and A. Billard, "On Learning, Representing, and Generalizing a Task in a Humanoid Robot," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 37, no. 2, pp. 286–298, 2007.

[12] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot Programming by Demonstration," in *Handbook of Robotics*, ch. 59, 2008.

[13] K. Kronander and A. Billard, "Online learning of varying stiffness through physical human-robot interaction," *2012 IEEE International Conference on Robotics and Automation*, pp. 1842–1849, May 2012.