Document downloaded from:

http://hdl.handle.net/10251/201738

This paper must be cited as:

Escarabajal-Sánchez, RJ.; Pulloquinga-Zapata, J.; Zamora-Ortiz, P.; Valera Fernández, Á.; Mata Amela, V.; Vallés Miquel, M. (2023). Imitation Learning-Based System for the Execution of Self-Paced Robotic-Assisted Passive Rehabilitation Exercises. IEEE Robotics and Automation Letters. 8(7):4283-4290. https://doi.org/10.1109/LRA.2023.3281884

The final publication is available at

https://doi.org/10.1109/LRA.2023.3281884

Copyright Institute of Electrical and Electronics Engineers

Additional Information

Imitation Learning-based System for the Execution of Self-paced Robotic-assisted Passive Rehabilitation Exercises

Rafael J. Escarabajal, José L. Pulloquinga, Pau Zamora-Ortiz, Ángel Valera, *Member, IEEE*,
Vicente Mata, and Marina Vallés

***Abstract*—The development of robotic-assisted rehabilitation exercises involving physical human-robot interaction requires extreme care since an injured limb may be in physical contact with the robot, so compliant behavior is imperative for these tasks. Typical approaches involve force control schemes like admittance controllers that allow humans to adapt the motion. However, when the patient's limb has limited mobility or is potentially injured, unintentional forces may occur during the robot's trajectory that could be incompatible with these controllers. This paper addresses a new way of generating compliant trajectories for passive rehabilitation exercises, considering that previous positions of the trajectory are attainable for the patient, so reversing the trajectory is a safe operation. Since there is no clear way to optimize such a goal due to the physiological variability among patients, the condition of reversal is based on imitation learning by taking the analogous healthy limb of the patient as a reference and encoding the forces using Gaussian Mixture Regression, and reversibility is accomplished by means of Reversible Dynamic Movement Primitives. The system allows for self-paced rehabilitation exercises by back-and-forth movements along the trajectory according to the patient's reaction, and it has been successfully applied to a 4-DOF parallel robot for lower-limb rehabilitation.**

***Index Terms*—Rehabilitation robotics, Learning from demonstrations, Reversible Dynamic Movement Primitives, Gaussian Mixture Regression, Parallel robot.**

I. INTRODUCTION

APPLICATIONS of robotics are evolving toward their use in non-structured environments such as households or hospitals. However, there is a big gap in domain knowledge, safety, and dependability that needs to be solved, especially in physical human-robot interaction settings, as stated in [1]. In the medical field, rehabilitation robotics is an emerging discipline that can involve either physical interaction or teleoperation, and [2] describes some advances in this topic.

Within the area of rehabilitation with physical human-robot interaction, many typologies of robots can be employed, such as exoskeletons [3], [4], and parallel robots (PRs) [5]. Likewise, many types of exercises can be performed with the assistance of robots, namely: active exercises, in which the robot and the patient exchange forces; and passive exercises, where the human needs to reach certain positions with no need of force exchange. There are also other variants in-between, such as active-assisted exercises [6].

Active exercises usually make use of force control [7], from which impedance and admittance-like controllers are the predominant ones for interaction between humans and other surfaces [8]. Passive exercises seem simpler in terms of control since they are meant for mobility augmentation by progressively reaching a goal position, so a position controller is apparently enough, and thus less literature is devoted to this topic (for an example, see [9]). However, reaching those positions may not be possible for the patient since the injury may reduce their motor skills, making it impossible to extend the limb beyond a limit position. In the attempt, the patient will likely exert unpredictable forces on the robot, both in magnitude and direction, which is incompatible with previously commented force controllers. There is an added problem related to the morphological variability among patients, which makes it hard to clearly define an objective suitable for all of them, even with the same impairment. This is why we resort to imitation learning to customize the goal to each patient.

This paper proposes the implementation of an algorithm for passive exercises accounting for unexpected forces due to the limited mobility of the patient. The method presented in this study performs the reversal of the trajectory when such forces from a human's limb occur. It can be used either for arms or legs and can be applied to any kind of robot that interacts with limbs. This setup assumes that positions experienced earlier in a trajectory are acceptable in terms of motor skills and effort for the patient, so they won't exert a considerable force on the robot in a regular regime until the trajectory leads the limb to a position incompatible with the injury. The baseline force for comparison is obtained via imitation learning from the analogous healthy limb, to extrapolate the features and allow customization. The trajectories are defined beforehand by a physiotherapist, so they are suitable for the rehabilitation.

A survey of imitation learning techniques can be found in

This work was supported in part by the Fondo Europeo de Desarrollo Regional under Grant PID2021-125694OB-I00, in part by the Vicerrectorado de Investigación de la Universitat Politècnica de València under Grant PAID-11-21, and in part by the Ministerio de Universidades, Gobierno de España under Grant FPU18/05105. *(Corresponding author: Rafael J. Escarabajal.)*

Rafael J. Escarabajal, José L. Pulloquinga, Ángel Valera and Marina Vallés are with the Departamento de Ingeniería de Sistemas y Automática, Instituto de Automática e Informática Industrial, Valencia, Camino de Vera, s/n, Spain (e-mail: raessan2@doctor.upv.es; jopulza@doctor.upv.es; giuprog@isa.upv.es; mvalles@isa.upv.es)

Pau Zamora-Ortiz is with the Instituto Universitario de Ingeniería Mecánica y Biomecánica, Universitat Politècnica de València, Valencia, Camino de Vera, s/n, Spain (e-mail: pazaor@upv.es)

Vicente Mata is with the Universitat Politècnica de València, Departamento de Ingeniería Mecánica y de Materiales, Centro de Investigación en Ingeniería Mecánica, Valencia, Camino de Vera, s/n, Spain (e-mail: vmata@mcm.upv.es)

Digital Object Identifier (DOI): see top of this page.

[10]. One of the leading techniques is Dynamic Movement Primitives (DMPs), which were pioneered by [11], [12] and advanced in [13]. They are inspired in biological systems and encode a reference signal using a set of differential equations with attractor properties and an adjustable nonlinear term.

The major extension of DMPs that is exploited in this paper is the possibility to add the reversibility property, which is a recent application that was first suggested by [14] but lacked asymptotic stability and reversibility, and [15] then improved the formulation. The last constitutes the basis of the reversibility mechanism proposed in this paper.

The trigger for reversing the trajectory in this work is defined by comparing the force exerted by the injured limb with a previously encoded threshold defined with the healthy limb, given as a baseline using Gaussian Mixture Regression (GMR) for the same exercise. This technique uses the properties of the multivariate gaussian distribution [16], [17].

The connection between the Reversible DMP (RDMP) and the GMR in this work is realized by means of the (reversible) phase variable of the RDMP, which replaces time and keeps the coordination during the exercise. When the limit encoded in the GMR is exceeded by the patient's force, the phase is reverted according to a heuristic function until the patient can relax the limb back and retry the exercise. This leads to self-paced and automatic progress of the patient. The application was tested on an actual lower-limb rehabilitation PR [18].

Therefore, the main contribution of this paper is the use of force feedback to intelligently decide the direction of the evolution of a predefined rehabilitation trajectory (i.e., backward or forward), and the velocity required to keep the exercise adjustable for the motor skills of the user.

After the related work in Section II, we present a background in Section III comprising the formulation of the DMP employed in our application and the GMR system. In Section IV, the robotic architecture and the combined system are explained and the experiment is defined. Section V shows the results and evaluation of the experiment. Finally, Section VI presents the conclusions.

II. RELATED WORK

Lower limb rehabilitation robotics is currently an active research area. [19] discusses the importance, frequency, and treatment of strokes from a robotic and control perspective.

The use of force feedback for changing the robot's position by imitating an impedance model was pioneered by [20]. Most works on rehabilitation robotics use force feedback for any variant of force-position control. An example is the Lokomat robot for gait restoration described in [21].

Passive exercises have been reported for ankle rehabilitation in [9] based on Iterative Learning Control (ILC) and a phase stopping mechanism of the DMP. Using an emergency button, they moved the robot to a rest position. However, the path to reach that position was defined automatically by the algorithm using a straight line, which may not be the best solution since the patient may experience new compromising positions along the path. Moreover, that stopping mechanism led to the end of the exercise, removing the possibility of resuming it and forcing them to start over. Freezing the position with an emergy stop is also not an option since that ending position is presumably inadmissible for the user.

Regarding other imitation learning techniques, [22] employ probabilistic methods like GMR in humanoid robots to define behaviors based on multiple demonstrations by extracting relevant features from tasks to generalize the learned skills. In our case, those features are extracted from several demonstrations of forces exerted by the healthy limb. Another similar method for learning features is Gaussian Process Regression (GPR), an example of which was described by [23], where they combined it with the DMP in tasks of reaching and grasping using vision.

III. THEORETICAL FOUNDATION

This Section provides an intuition about the two coexisting mechanisms: RDMP and GPR. The former encodes the trajectory tracked by the robot and allows flexible manipulation of the trajectory by means of the phase variable. The latter encodes the force profile of the healthy limb along different demonstrations and allows establishing statistical criteria to make decisions regarding the evolution of the trajectory.

A. Reversible Dynamic Movement Primitives

The attractor dynamics present in conventional Dynamic Movement Primitives [13] prevents the system from achieving reversibility. [15] proposed a new formulation based on a linear system with a global asymptotically stable origin that can be expressed for 1 degree of freedom (DOF) as follows:

$$\ddot{y} = \ddot{y}_x - D(\dot{y} - \dot{y}_x) - K(y - y_x) \tag{1}$$

where K, D define the dynamics of the system so that it is overdamped or critically damped, and y_x is the desired scaled trajectory that is learned:

$$y_x = k_s(f_p(x) - f_p(x_0)) + y_0 \tag{2}$$

where k_s is the scaling factor defined as:

$$k_s = \frac{g - y_0}{f_p(x_f) - f_p(x_0)} \tag{3}$$

The trajectory evolves from the initial position y_0 to the goal position g according to a canonical system governed by the variable phase x, which monotonically decreases from its initial value $x(0) = x_0$ to a final value $x(t_f) = x_f$. The canonical system is usually selected as a first-order system:

$$\tau\dot{x} = -\alpha_x x \tag{4}$$

with τ the temporal factor equal to the temporal length of the trajectory, and α_x the positive time constant or exponential decay. If $x(0) = 1$, the phase tends to 0 exponentially.

The term $f_p(x)$ encodes the scaled trajectory through a linear combination of N radial basis functions dependent on the phase as follows:

$$f_p(x) = \frac{\sum_{i=1}^{N} w_i \Psi_i(x)}{\sum_{i=1}^{N} \Psi_i(x)} \tag{5}$$

$$\Psi_i(x) = \exp(-h_i(x - c_i)^2) \tag{6}$$

where c_i are the centers of the radial basis functions distributed along the canonical system, and h_i the bandwidths. The weights w_i are learned to best fit equation (2), usually employing regression techniques such as Locally Weighted Regression (LWR), [24]. The derivatives of the phase-dependent desired positions are obtained through equation (2): $\dot{y}_x = k_s \dot{f}_p(x)$, $\ddot{y}_x = k_s \ddot{f}_p(x)$, with:

$$\dot{f}_p(x) = \frac{\partial f_p}{\partial x}\dot{x} \tag{7}$$

$$\ddot{f}_p(x) = \frac{\partial^2 f_p}{\partial x^2}\dot{x}^2 + \frac{\partial f_p}{\partial x}\ddot{x} \tag{8}$$

After learning the weights, the system can be set in motion by integration of equations (1,4) from their initial conditions. By default, the canonical system monotonically decreases, reproducing the forward trajectory. Some existing mechanisms slow down the trajectory during integration [25]. In this paper, we propose a new slight modification in the canonical system during integration by defining a relative velocity variable $\nu \in [-1, 1]$ leaving the canonical system as:

$$\tau\dot{x} = -\nu\alpha_x x \tag{9}$$

The concept of relative velocity comes from the fact that leaving $\nu = 1$ makes the system evolve with the default velocity (sampled from the references file at a given frequency), and decreasing ν in absolute value decreases the velocity as a fraction of the nominal. A negative relative velocity makes the phase evolve in the opposite direction, and so does the trajectory by virtue of the reversibility capabilities of this formulation. Therefore, if $\nu = -1$, the trajectory goes backward at nominal velocity. Theoretically, the range of ν can be of choice, but in our study, the nominal velocity will not be exceeded. This novel approach offers much more control than the slow-down mechanism.

With this new formulation, special care must be taken since now the second derivative $\ddot{x}$ to plug in equation (8) must take this new time-dependent variable into account:

$$\ddot{x} = -\frac{\alpha_x}{\tau}(\nu\dot{x} + \dot{\nu}x) \tag{10}$$

In this study, we obtain $\dot{\nu}$ as the discretized derivative by calculating the increment between two samples: $\dot{\nu} \approx \frac{\nu_k - \nu_{k-1}}{T_s}$, where T_s is the sample time of the control system.

The phase x should also be clipped to the allowed range $x \in [x_0, x_f]$ such that it lies within the valid region of the definition of the trajectory. For several DOFs, each of them has its associated DMP but shares the common phase and the canonical system to allow synchronization.

By choosing a reasonable law for ν (without severe jumps), the phase evolves without discontinuities, and stability is ensured as described in [15]. The rest of the study focuses on the definition of ν based on the forces exchanged between the user and the robot.

B. Gaussian Mixture Regression

Gaussian mixture regression (GMR) is a technique that uses the well-known properties of multivariate gaussian distributions to perform a probabilistic regression in an indirect way: first, it encodes both inputs and outputs together with a Gaussian mixture model (GMM), and then it performs the regression fitting the model to the outputs using the conditioning property of the gaussian distribution [16].

Given time-distributed input and output datapoints $\mathbf{s_t^I}, \mathbf{s_t^O}$, the data can be characterized by centers $\boldsymbol{\mu}_\mathbf{i}$ and covariances $\boldsymbol{\Sigma}_\mathbf{i}$, with $i = 1 \dots M$ and M the number of gaussians:

$$\mathbf{s_t} = \begin{bmatrix} \mathbf{s_t^I} \\ \mathbf{s_t^O} \end{bmatrix} \quad \boldsymbol{\mu}_\mathbf{i} = \begin{bmatrix} \boldsymbol{\mu}_\mathbf{i}^\mathbf{I} \\ \boldsymbol{\mu}_\mathbf{i}^\mathbf{O} \end{bmatrix} \quad \boldsymbol{\Sigma}_\mathbf{i} = \begin{bmatrix} \boldsymbol{\Sigma}_\mathbf{i}^\mathbf{I} & \boldsymbol{\Sigma}_\mathbf{i}^\mathbf{IO} \\ \boldsymbol{\Sigma}_\mathbf{i}^\mathbf{OI} & \boldsymbol{\Sigma}_\mathbf{i}^\mathbf{O} \end{bmatrix} \tag{11}$$

After obtaining the probability density function $\mathcal{P}(\mathbf{s_t})$ with the GMM [26], the model is fed with new incoming input data $\mathbf{s_t^I}$ to compute the multimodal conditional distribution $\mathcal{P}(\mathbf{s_t^O}|\mathbf{s_t^I})$ as follows:

$$\mathcal{P}(\mathbf{s_t^O}|\mathbf{s_t^I}) = \sum_{i=1}^{M} h_i \mathcal{N}(\mathbf{s_i^O}|\hat{\boldsymbol{\mu}}_\mathbf{i}^\mathbf{O}, \hat{\boldsymbol{\Sigma}}_\mathbf{i}^\mathbf{O}) \tag{12}$$

$$\hat{\boldsymbol{\mu}}_\mathbf{i}^\mathbf{O} = \boldsymbol{\mu}_\mathbf{i}^\mathbf{O} + \boldsymbol{\Sigma}_\mathbf{i}^\mathbf{OI} {\boldsymbol{\Sigma}_\mathbf{i}^\mathbf{I}}^{-1}(\mathbf{s_t^I} - \boldsymbol{\mu}_\mathbf{i}^\mathbf{I}) \tag{13}$$

$$\hat{\boldsymbol{\Sigma}}_\mathbf{i}^\mathbf{O} = \boldsymbol{\Sigma}_\mathbf{i}^\mathbf{O} - \boldsymbol{\Sigma}_\mathbf{i}^\mathbf{OI} {\boldsymbol{\Sigma}_\mathbf{i}^\mathbf{I}}^{-1} \boldsymbol{\Sigma}_\mathbf{i}^\mathbf{IO} \tag{14}$$

$$h_i = \frac{\pi_i \mathcal{N}(\mathbf{s_t^I}|\boldsymbol{\mu}_\mathbf{i}^\mathbf{I}, \boldsymbol{\Sigma}_\mathbf{i}^\mathbf{I})}{\sum_{m=1}^{M} \pi_m \mathcal{N}(\mathbf{s_t^I}|\boldsymbol{\mu}_\mathbf{m}^\mathbf{I}, \boldsymbol{\Sigma}_\mathbf{m}^\mathbf{I})} \tag{15}$$

where $\mathcal{N}(\mathbf{s_t^I}|\boldsymbol{\mu}_\mathbf{i}^\mathbf{I}, \boldsymbol{\Sigma}_\mathbf{i}^\mathbf{I})$ is the multivariate normal distribution:

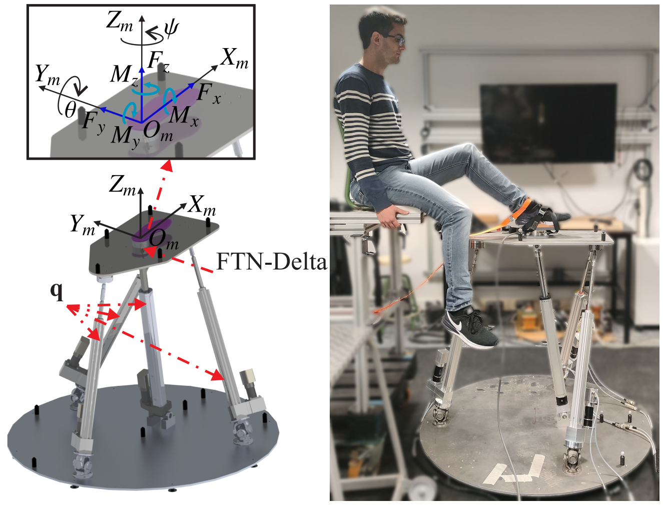
$$\begin{aligned} \mathcal{N}(\mathbf{s_t^I}|\boldsymbol{\mu}_\mathbf{i}^\mathbf{I}, \boldsymbol{\Sigma}_\mathbf{i}^\mathbf{I}) = {} & (2\pi)^{-\frac{L}{2}} |\boldsymbol{\Sigma}_\mathbf{i}^\mathbf{I}|^{-\frac{1}{2}} \\ & \exp\left(-\frac{1}{2}(\mathbf{s_i^I} - \boldsymbol{\mu}_\mathbf{i}^\mathbf{I})^\mathbf{T} {\boldsymbol{\Sigma}_\mathbf{i}^\mathbf{I}}^{-1} (\mathbf{s_t^I} - \boldsymbol{\mu}_\mathbf{i}^\mathbf{I})\right) \end{aligned} \tag{16}$$

with L defining the dimension of the gaussian distribution, and π_i are the mixing coefficients from the GMM. According to equations (12-15), the output is characterized using M means and covariances. Nevertheless, the use of a unimodal approximation $\mathcal{P}(\mathbf{s_t^O}|\mathbf{s_t^I}) = \mathcal{N}(\mathbf{s_t^O}|\hat{\boldsymbol{\mu}}_\mathbf{t}^\mathbf{O}, \hat{\boldsymbol{\Sigma}}_\mathbf{t}^\mathbf{O})$ gives a better understanding and can be realized employing the law of total mean and variance:

$$\hat{\boldsymbol{\mu}}_\mathbf{t}^\mathbf{O} = \sum_{i=1}^{M} h_i \hat{\boldsymbol{\mu}}_\mathbf{i}^\mathbf{O} \tag{17}$$

$$\hat{\boldsymbol{\Sigma}}_\mathbf{t}^\mathbf{O} = \sum_{i=1}^{M} h_i (\hat{\boldsymbol{\Sigma}}_\mathbf{i}^\mathbf{O} + \hat{\boldsymbol{\mu}}_\mathbf{i}^\mathbf{O} \hat{\boldsymbol{\mu}}_\mathbf{i}^{\mathbf{O}^\mathbf{T}}) - \hat{\boldsymbol{\mu}}_\mathbf{t}^\mathbf{O} \hat{\boldsymbol{\mu}}_\mathbf{t}^{\mathbf{O}^\mathbf{T}} \tag{18}$$

The choice of GMR over GPR lies in the way they interpret the variance: while GMR contemplates the variability in the training data, GPR measures the degree of uncertainty, i.e., the presence or absence of data in the region around the query point (see [16]). In this study, we collect different demonstrations with shared input data, and the variability among the outputs is processed, so GMR is more suitable.



(a) (b)

Fig. 1: (a) PR for rehabilitation and force sensor, and (b) experimental setup.

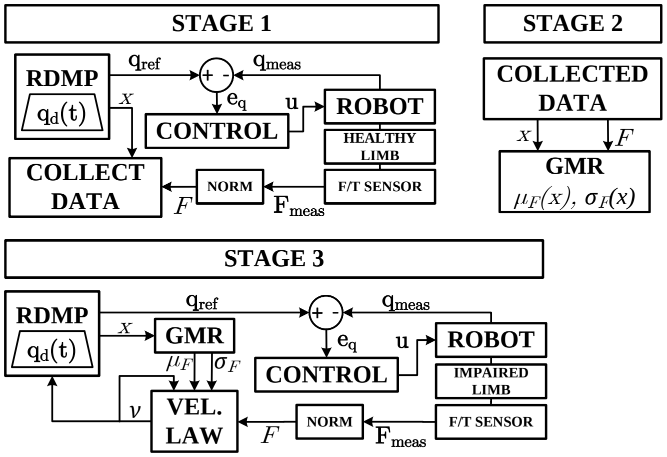


Fig. 2: Parts of the experiment. Stage 1: demonstrations using the healthy limb. Stage 2: offline data processing. Stage 3: performance using the limb with mobility constraints.

IV. SELF-PACED PASSIVE REHABILITATION SYSTEM

This Section describes the 4-DOF PR and the proposed mechanism combining the previous ideas to perform self-paced passive rehabilitation exercises.

A. 4-DOF parallel robot for lower-limb rehabilitation

The 4-DOF PR was designed at the Universitat Politècnica de València [18] for lower-limb rehabilitation, involving tasks such as flexion-extension, internal-external rotation, and hip flexion. Its four DOFs include two translational movements (X_m, Z_m) in the tibiofemoral plane, one rotation (ψ) around the coronal plane, and one rotation (θ) around the tibiofemoral plane (see Figure 1a).

Its structure is $3U\underline{P}S$+$R\underline{P}U$, where the letters U, P, S, and R stand for universal, prismatic, spherical, and revolute joints, respectively, and the underline format indicates the actuated joints. Therefore, the connection between the mobile and fixed platform involves three external limbs in $U\underline{P}S$ configuration and a central limb in $R\underline{P}U$ configuration. The four actuated (prismatic) joints are collected in a vector $\mathbf{q}$.

The interaction between the user and the robot is measured with an FTN-Delta sensor manufactured by Schunk, which is attached on top of the mobile platform. It provides six-axis force/torque measurements, and its streaming data is sent via UDP. The user's foot can be attached to the sensor using a boot or some straps. Figure 1b shows the complete system. Since the experiments are performed with a healthy user, the orange strap allows the simulation of mobility constraints by exerting external forces, affecting the passive exercise.

The system is implemented in an industrial computer using Robot Operating System 2 (ROS2) and the C++ programming language, with a sample time of $T_s = 10\,\mathrm{ms}$.

B. Description of the system

This work proposes a novel mechanism to perform passive rehabilitation exercises involving a lower limb with potentially reduced mobility. The robot is aware of such reduced mobility by comparison with the performance using the analogous healthy limb through several demonstrations. To this end, the exercise is divided into three stages (see Figure 2). The input is a desired trajectory in cartesian coordinates, designed with the help of a physiotherapist:

$$\mathbf{p_d} = [X_{mt}, Z_{mt}, \theta_{mt}, \psi_{mt}]_{t=1}^{T} \tag{19}$$

where $t = 1 \dots T$ are the samples. The cartesian positions can be converted to joint coordinates: $\mathbf{q_d} = IK(\mathbf{p_d})$, where $IK(\cdot)$ is the Inverse Kinematics operator.

1) Stage 1: The first step involves performing the passive exercise with the healthy limb. Being passive does not mean that the interaction forces are zero, since the resting limb attached to the robot exchanges forces (e.g., gravity) depending on the position of the platform. Moreover, even with the limb relaxed, the interaction forces from one iteration to the next change since muscle relaxation is also subject to variability, provoking the appearance of residual forces that can depend on the position, but also vary from one patient to another. To capture this force variability, n demonstrations are employed.

In this stage, the joint trajectory $\mathbf{q_d}$ is first encoded with the RDMP, and then it outputs the current phase x and reference to follow $\mathbf{q_{ref}}(x)$ (note that we make the distinction between the initial desired trajectory $\mathbf{q_d}$ from the file and the actual trajectory that is tracked $\mathbf{q_{ref}}$, subject to change by the RDMP). The position tracking is performed by a simple joint PID control, which accepts the reference and measured positions ($\mathbf{q_{meas}}$) and outputs the control action.

During this stage, the trajectory is not reversed, and the measured force is collected, composed of six terms expressed in cartesian coordinates with respect to the mobile platform: $\mathbf{F_{meas}} = [F_x, F_y, F_z, M_x, M_y, M_z]^T$ (blue vectors in Figure 1a). These are calibrated (set to 0) at the beginning of each experiment to avoid drifts. From these data, we establish a scalar indicator of the force F as follows:

i) The torques are mapped to forces by using equivalent distances l_{eq_x}, l_{eq_y}, l_{eq_z} which, according to Figure 3,

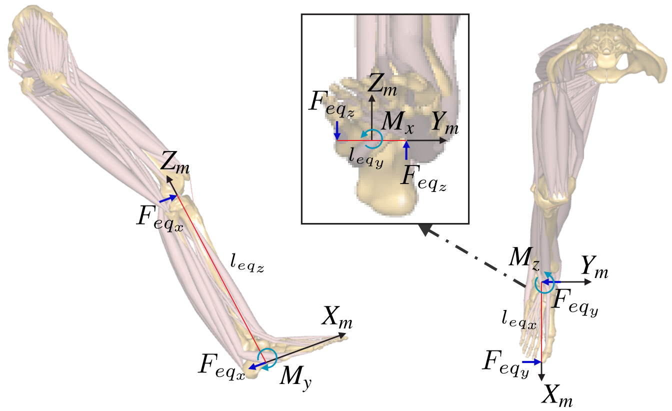


Fig. 3: Mapping of measured torques to equivalent forces in each axis. Lengths are in red, torques are in cyan, equivalent forces are in dark blue.

are equal to the length of the foot, the width of the foot, and the length of the tibia, respectively. These allow us to obtain the equivalent forces due to the torques: $F_{eq_x} = \frac{M_y}{l_{eq_z}}$, $F_{eq_y} = \frac{M_z}{l_{eq_x}}$, and $F_{eq_z} = \frac{M_x}{l_{eq_y}}$.

ii) The norm of the new vector of forces $F = ||[F_x, F_y, F_z, F_{eq_x}, F_{eq_y}, F_{eq_z}]^T||$ is calculated and selected as a representative scalar of the overall force.

The use of the norm in calculating the scalar representative force makes the algorithm robust to movements or forces that are symmetric but opposite in direction between the healthy and injured limb, such as the rotation ψ around the coronal plane. Additionally, the sensor calibration at the beginning of the experiment disregards the gravitational force of the limb, providing extra robustness in cases of little variations in limb weight. Together, these measures ensure the valid comparison of the representative forces of both limbs.

Using the norm has another benefit, since it reduces all force components into a single effect that is straightforward to understand and implement (as opposed to considering all individual forces separately). This simplicity is supported by the fact that we are dealing with passive exercises, which allows the design of an algorithm that can accurately distinguish between normal and abnormal forces using this metric.

On the other hand, the conversion from torques to forces in the first step allows us to create a standardized set of variables with consistent units. This way, we are able to work with a single variable (force) throughout the entire analysis.

The phase x of the RDMP and the representative force F of the n demonstrations are stored for processing in Stage 2.

2) Stage 2: This offline step builds a GMR model with M gaussians using the collected data. Traditionally, the input to the GMR is the time variable. However, this would fail since, in the case of trajectory reversal, time would keep evolving forward while the trajectory progresses backward, leading to the desynchronization between the position and the corresponding force outputted by the GMR.

To achieve the desired synchronization, the encoded input is the phase x, so $\mathbf{s_t^I} = x$, $\mathbf{s_t^O} = F$. After the training is performed according to equations (12-18), the model is able to output the mean $\hat{\boldsymbol{\mu}}_\mathbf{t}^\mathbf{O} = \mu_F$ and standard deviation $\hat{\boldsymbol{\Sigma}}_\mathbf{t}^\mathbf{O} = \sigma_F$ when fed with the phase.

The GMR does not include the velocity as an input because the velocities used in rehabilitation with robotic devices are typically not high enough to require consideration of velocity-dependent or inertial forces. Moreover, including velocity as an input would require an increase in the number of demonstrations to compensate for the added dimension.

3) Stage 3: This is the final stage that is performed with the impaired limb. The RDMP works with the same trajectory and outputs a joint reference to be tracked by the robot as before, but this time it receives the relative velocity as input ν, which allows it to control the evolution of the trajectory. The definition of ν depends on:

i) The output of the learned GMR model, (i.e., the expected force μ_F and its uncertainty σ_F).
ii) The force read from the sensors, which is processed as in Stage 1 to obtain the representative force F.
iii) The previous value of the output ν_{k-1}.

The law to define ν_k is based on the calculation of the z-score, which reveals its distance from the mean, measured in standard deviation units:

$$z_F = \frac{F - \mu_F}{\sigma_F} \tag{20}$$

The algorithm sets a threshold for this number z_{lim} and calculates the output ν_k with the following law:

$$\nu_k = \begin{cases} max(\nu_{k-1} - c_1\nu_{k-1}^2 - \\ \qquad c_2(z_F - z_{lim}), -1), & \text{if } |z_F| \geq z_{lim} \\ min(\nu_{k-1} + c_3\nu_{k-1}^2 + c_4, 1), & \text{otherwise} \end{cases} \tag{21}$$

The first case occurs when the force exerted by the impaired limb is abnormally different from those of the demonstrations. Then, the relative velocity decreases (clipped to -1, corresponding to complete reversal) quadratically with respect to the previous velocity, weighted with the coefficient c_1, and linearly with respect to the excess of force with coefficient c_2. When the force is within the limit (second case) ν tends to increase up to a maximum of 1, but now the term related to the force disappears because it fulfills the threshold, leaving a constant c_4. The constants are positive.

We have chosen to use the quadratic term because we want the robot to have a slow velocity during the transitions, like when it switches from moving backward to moving forward. This gives the user a few seconds to adapt. The sensitivity z_{lim} is also variable to adjust the trigger of the mechanism.

Our algorithm does not set a limit on the duration of the exercise. Instead, it allows the physiotherapist to determine the appropriate endpoint for each session based on the patient's individual needs and progress. This endpoint can be based on various factors such as the patient's physical capacity, level of fatigue, degree of completion of the exercise, or pain tolerance. This procedure also offers a self-paced passive rehabilitation exercise since the user can go back and forth as many times as needed without external intervention.

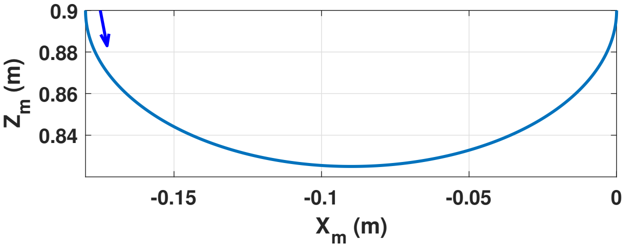


Fig. 4: Cartesian coordinates of the knee extension exercise. The arrow indicates the starting point and evolution of the exercise.

C. *Trajectory and exercise*

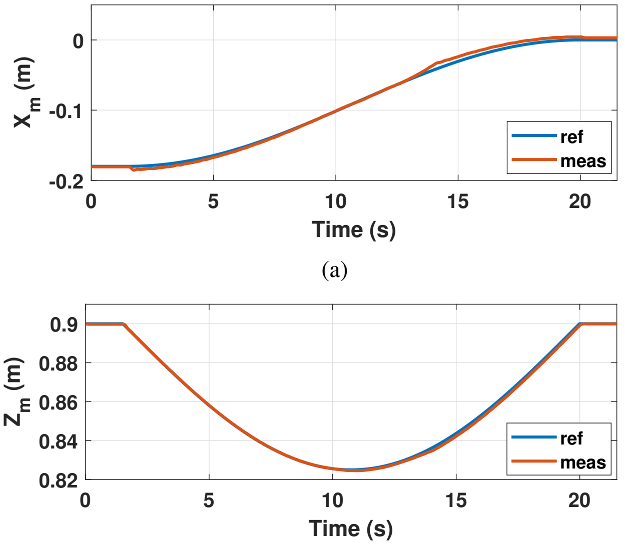
The chosen exercise for this experiment is a knee extension in the tibio-femoral plane, which uses the translations X_m and Z_m and has a semicircumferential shape, starting at $[X_m, Z_m] = [-0.18, 0.9]^T$ and finishing at $[X_m, Z_m] = [0.0, 0.9]^T$ (see Figure 4). The orientations θ and ψ are 0. The nominal velocity is $0.015\,\frac{m}{s}$, and the total duration of the trajectory is $21.5\,\mathrm{s}$.

The trajectory is encoded using $n = 5$ demonstrations with a GMR model with $M = 8$ states. These values were selected empirically according to the duration of the trajectory and the variability of the demonstrations. For the reproduction with the impaired limb, we set the value of z_{lim} to 5 to ensure that the robotic device changes direction only when the force is significantly abnormal. We first determined the appropriate order of magnitude based on the concept of the z-score applied to a pure gaussian distribution. However, since our method involves a combination of gaussians, we then extrapolated this concept and empirically tuned the value through several experiments with the healthy limb, ensuring that the robot did not change direction unexpectedly. This approach strikes a good tradeoff by ensuring that the trajectory is not reversed during normal conditions but the robot responds promptly when the force becomes abnormal.

The value of ν_0 was set to 1, indicating that it starts with the nominal velocity, and the constants for the ν_k law are $c_1 = 0.05, c_2 = 0.01, c_3 = 0.005, c_4 = 0.001$. Note that the constants c_1 and c_2 are 10 times higher than c_3 and c_4. The intended effect is to make the dynamics of the transition from forward to backward movement faster than the opposite situation. This is because when the robot is going forward and the force exceeds the limit, the robot should quickly change the direction since the position may not be compatible with the patient's motor skills. On the contrary, when the robot changes from backward to forward motion, the slow dynamics allows the patient to space out the repetitions.

Regarding the parameters of the RDMP, the spring coefficient (K) and damping (D) were assigned values of 1.0 and 2.0, respectively, making the system critically damped. The time constant of the canonical system α_x was set to 2.0. The chosen number of basis functions to accurately encode the trajectory is $N = 1000$. The centers of the basis functions were defined as $c_i = \exp(-\alpha_x \frac{i-1}{N-1})$. The bandwidths h_i were constant for all the basis functions, with a value of 0.75. We provide unitless values since it is a virtual system.

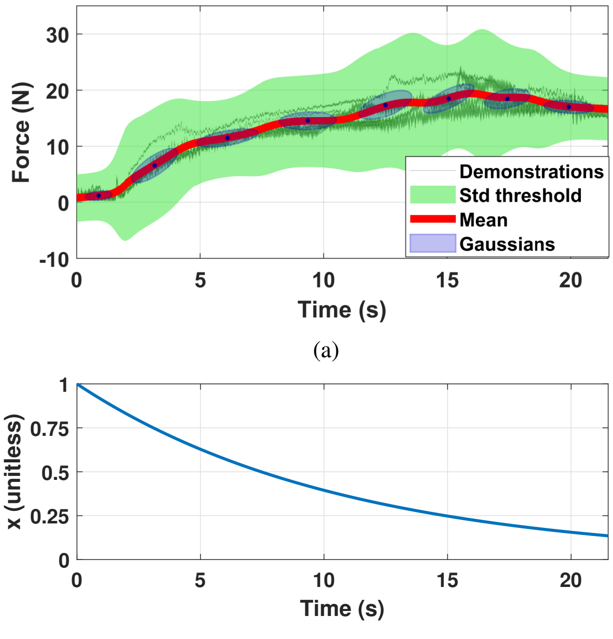
The experiment was carried out with one healthy male subject, who was informed about the experiment and provided his consent. He performed Stage 1 of the experiment with the right lower limb, and Stage 3 with the left limb, with a simulated muscular impairment using a strap (Figure 1b).



(a)

(b)

Fig. 5: Original trajectory expressed in temporal domain for the coordinates a) X_m and b) Z_m.



(a)

(b)

Fig. 6: a) Force demonstrations during stage 1 and result of the predictions using GMR (in terms of mean and standard deviation). The input is the phase x, plotted in b).

V. RESULTS AND EVALUATION

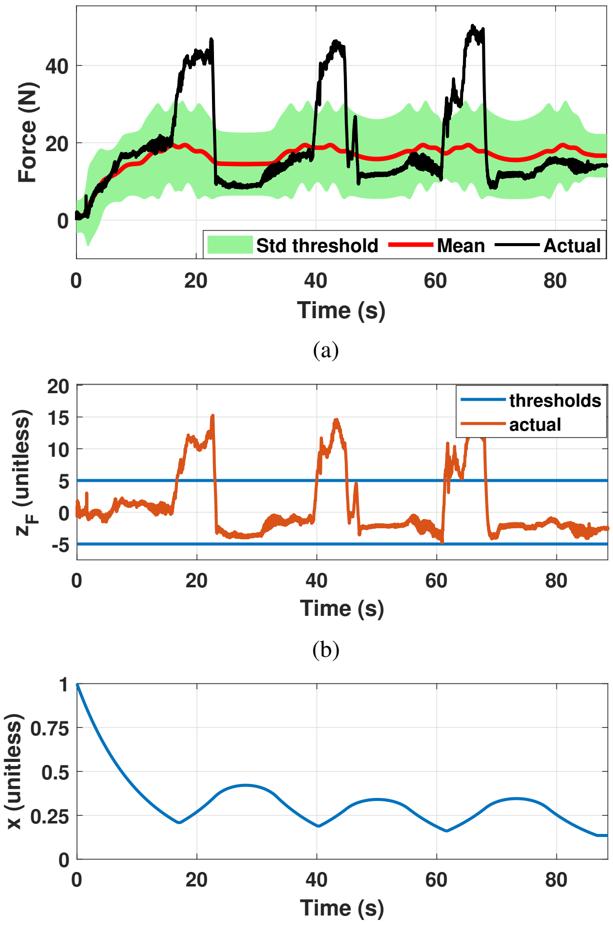
Figure 5 depicts the cartesian reference and measured position of the translational coordinates through time during the initial demonstrations of Stage 1 (the plot corresponds to one of them), so they are not altered and evolve according to the reference file to produce the spatial trajectory of Figure 4.

The forces and the phase of the system are collected for the 5 demonstrations. Figure 6a represents the value of F for all the demonstrations and the mapped μ_F and threshold using σ_F. Specifically, the green shaded region corresponds to the separation of the amount $z_{lim}\sigma_F$ from the mean, and expresses the limit of exerted force tolerated before changing the direction. The estimated gaussians of the underlying GMM model are also depicted, where the shape and orientation of the ellipses are defined by the variances and covariances of the input phase x and output force F jointly. The clustering of the gaussians allows for the probabilistic prediction of forces after learning the model, by combining the effect of all the gaussians according to equations (12-18). The centers are dependent on the phase x, but they have been mapped to the time domain for visualization purposes. The result shows that the 8 gaussians are approximately equidistant in time to better capture the variability of the force along the complete interval. The phase x used as input by the GMR is also depicted in Figure 6b. It exhibits the exponential behavior from the first-order system.

After learning the model, the trajectory is performed with the impaired limb. Using this setup, the user met mobility restrictions in several stages of the trajectory, and a total of three attempts were performed in which the user exerted force greater than the limit (Figure 7a), activating the reversal mechanism. Figure 7b shows the force exceedance in terms of the z-score z_F, which reaches values around 15 during the execution. Finally, Figure 7c plots the evolution of the phase, which gets reversed when the force goes out of range.

The reversal mechanism becomes clearer observing the cartesian coordinates along the new trajectory, which are depicted in Figures 8a-b, where the new X_m and Z_m are plotted (both references and measurements) together with the changes in direction caused by the forces exchanged during the trajectory. The black vertical lines delimit these changes.

Figure 8c shows the effect of the relative velocity ν on the system. Starting from a value of 1 (forward evolution with nominal velocity), at instant $t = 16\,\text{s}$ (when the force

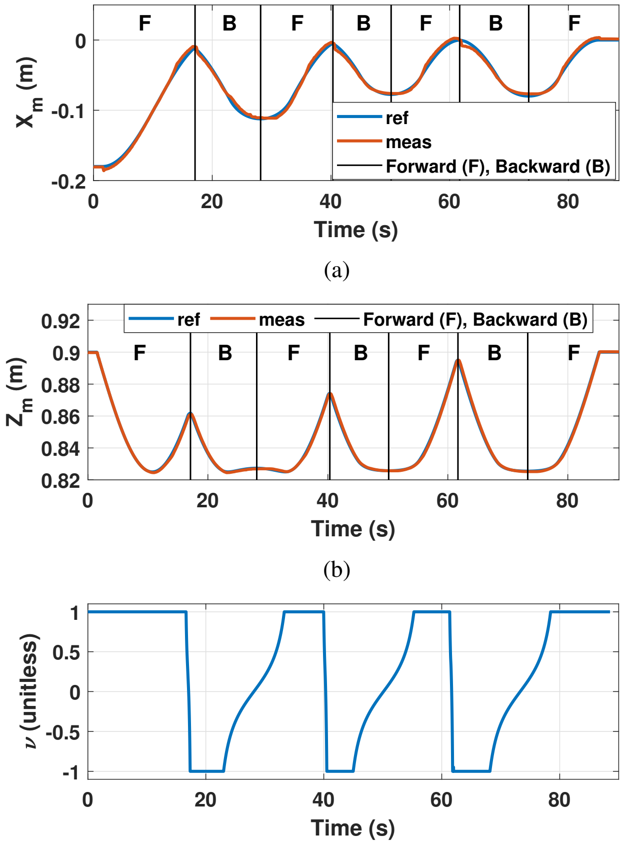


(a)

(b)

(c)

Fig. 7: a) Evolution of the exerted force in the experiment of Stage 3, and comparison with the output of the GMR model. b) Calculation of the z-score z_F, together with the thresholds. c) Evolution of the phase resulting from the exerted forces.



(a)

(b)

(c)

Fig. 8: Evolution of the cartesian references and measured positions a) X_m and b) Z_m for the exercise of Stage 3 with the corresponding progress (forward or backward), and c) relative velocity causing the inversion.

exceeds the threshold according to Figure 7) the velocity drops immediately down to -1 in just one second, obeying the law of equation (21). This allows a quick trajectory reversal to meet the user mobility requirements, and this immediate effect can be seen in the transitions F → B of Figures 8a-b. The transition B → F, however, is much slower and it starts when the patient recovers during the backward motion, which is detected by a decrease in the exerted force to lie within the limits. The convenient quadratic effect of equation (21) is clearly visible since the robot tends to slow down for a larger time interval around the point of null velocity, giving the user some time until next attempt.

A video with the execution of this exercise can be found in the URL: http://roboprop.ai2.upv.es/wp-content/uploads/2023/02/DMP_retroceso.mp4

VI. Conclusion

This paper presented a novel approach to performing passive rehabilitation exercises that provides the flexibility to execute self-paced compliant trajectories, where the compliance comes from the possibility of reversing the trajectory to keep the movements compatible with the user's motor skills. This methodology is robot-agnostic as long as it interacts with human limbs, with straightforward extrapolation.

The reversibility was achieved through the RDMP, which inherently incorporates this mechanism. In this study, we worked with the novel concept of relative velocity applied to the RDMP, which allows controlling the desired velocity, and the rest of the work focused on how to define it.

The chosen approach involves the comparison of the force exerted with the impaired limb with a customized baseline for each patient. This baseline comes from the execution of the same exercise with the analogous healthy limb, which allows extracting features regarding the variability through several force demonstrations and the encoding of the data using GMR, which is linked to the RDMP through its phase.

We also discussed the selection of the relevant parameters that characterize the reversal to achieve the intended behavior, and the results show how the system successfully reacts to the mobility restrictions of the user during the exercise.

In future research, some other features can be implemented for diagnosis. For example, a score based on the produced data can be designed to assess the evolution of the patient, together with other unused signals such as electromyography. Also, the thresholds and parameters of our algorithm can turn variable among attempts or experiments to account for the changes in the performance of the patient. Finally, an emergency button held by the patient or the physiotherapist can be added to our system for increased safety during rehabilitation, reversing the trajectory in case of any misbehavior.

References

[1] A. De Santis, B. Siciliano, A. De Luca, and A. Bicchi, "An atlas of physical human–robot interaction," *Mechanism and Machine Theory*, vol. 43, no. 3, pp. 253–270, 2008.

[2] S. Xie, *Advanced Robotics for Medical Rehabilitation: Current State of the Art and Recent Advances*, 1st ed. Springer, 2016.

[3] G. Chen, C. K. Chan, Z. Guo, and H. Yu, "A review of lower extremity assistive robotic exoskeletons in rehabilitation therapy," *Critical Reviews™ in Biomedical Engineering*, vol. 41, no. 4-5, 2013.

[4] D. Shi, W. Zhang, W. Zhang, and X. Ding, "A review on lower limb rehabilitation exoskeleton robots," *Chinese Journal of Mechanical Engineering*, vol. 32, no. 1, pp. 1–11, 2019.

[5] G. Liu, J. Gao, H. Yue, X. Zhang, and G. Lu, "Design and kinematics analysis of parallel robots for ankle rehabilitation," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 253–258.

[6] E. Akdoğan and M. A. Adli, "The design and control of a therapeutic exercise robot for lower limb rehabilitation: Physiotherabot," *Mechatronics*, vol. 21, no. 3, pp. 509–522, 2011.

[7] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Force Control*. London: Springer London, 2009, pp. 363–405.

[8] A. Q. Keemink, H. van der Kooij, and A. H. Stienen, "Admittance control for physical human–robot interaction," *The International Journal of Robotics Research*, vol. 37, no. 11, pp. 1421–1444, 2018.

[9] F. J. Abu-Dakka, A. Valera, J. A. Escalera, M. Abderrahim, A. Page, and V. Mata, "Passive exercise adaptation for ankle rehabilitation based on learning control framework," *Sensors*, vol. 20, no. 21, pp. 1–23, 2020.

[10] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne, "Imitation learning: A survey of learning methods," *ACM Comput. Surv.*, vol. 50, no. 2, pp. 1–35, apr 2017.

[11] A. Ijspeert, J. Nakanishi, and S. Schaal, "Learning rhythmic movements by demonstration using nonlinear oscillators," vol. 1, 01 2002.

[12] ——, "Movement imitation with nonlinear dynamical systems in humanoid robots," in *Proceedings 2002 IEEE International Conference on Robotics and Automation*, vol. 2, 2002, pp. 1398–1403.

[13] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical Movement Primitives: Learning attractor models for motor behaviors," *Neural Computation*, vol. 25, no. 2, pp. 328–373, 2013.

[14] I. Iturrate, C. Sloth, A. Kramberger, H. G. Petersen, E. H. Østergaard, and T. R. Savarimuthu, "Towards Reversible Dynamic Movement Primitives," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 5063–5070.

[15] A. Sidiropoulos and Z. Doulgeri, "A Reversible Dynamic Movement Primitive formulation," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 3147–3153.

[16] S. Calinon and D. Lee, "Learning control," in *Humanoid Robotics: a Reference*, P. Vadakkepat and A. Goswami, Eds. Springer, 2019, pp. 1261–1312.

[17] S. Calinon, "Mixture models for the analysis, edition, and synthesis of continuous time series," in *Mixture Models and Applications*, N. Bouguila and W. Fan, Eds. Springer, Cham, 2019, pp. 39–57.

[18] M. Vallés, P. Araujo-Gómez, V. Mata, A. Valera, M. Díaz-Rodríguez, Álvaro Page, and N. M. Farhat, "Mechatronic design, experimental setup, and control architecture design of a novel 4 DoF parallel manipulator," *Mechanics Based Design of Structures and Machines*, vol. 46, no. 4, pp. 425–439, 2018.

[19] X. Zhang, Z. Yue, and J. Wang, "Robotics in lower-limb rehabilitation after stroke," *Behavioural neurology*, vol. 2017, 2017.

[20] N. Hogan, "Impedance Control: An Approach to Manipulation," in *1984 American Control Conference*, 1984, pp. 304–313.

[21] M. Bernhardt, M. Frey, G. Colombo, and R. Riener, "Hybrid force-position control yields cooperative behaviour of the rehabilitation robot LOKOMAT," in *9th International Conference on Rehabilitation Robotics, 2005. ICORR 2005.*, 2005, pp. 536–539.

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

[23] D. Forte, A. Ude, and A. Kos, "Robot learning by Gaussian process regression," in *19th International Workshop on Robotics in Alpe-Adria-Danube Region (RAAD 2010)*, 07 2010, pp. 303 – 308.

[24] C. G. Atkeson, A. W. Moore, and S. Schaal, "Locally weighted learning," *Lazy learning*, pp. 11–73, 1997.

[25] S. Schaal, P. Mohajerian, and A. Ijspeert, "Dynamics systems vs. optimal control — a unifying view," in *Computational Neuroscience: Theoretical Insights into Brain Function*, ser. Progress in Brain Research, P. Cisek, T. Drew, and J. F. Kalaska, Eds. Elsevier, 2007, vol. 165, pp. 425–445.

[26] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*. Springer, 2006, vol. 4, no. 4.