Gesteme-free context-aware adaptation of robot behavior in human-robot cooperation

Federico Nessi[a,1], Elisa Beretta[a,b], Cecilia Gatti[a], Giancarlo Ferrigno[a], Elena De Momi[a]

[a] *Department of Electronics, Information and Bioengineering, Politecnico di Milano, p.zza Leonardo da Vinci, 32, 20133, Milano (Italy)*
[b] *Kuka Roboter GmbH, Zugspitzstrasse, 140, 86165, Augsburg (Germany)*

Abstract

Background. Cooperative robotics is receiving greater acceptance because the typical advantages provided by manipulators are combined with an intuitive usage. In particular, hands-on robotics may benefit from the adaptation of the assistant behavior with respect to the activity currently performed by the user. A fast and reliable classification of human activities is required, as well as strategies to smoothly modify the control of the manipulator. In this scenario, gesteme-based motion classification is inadequate because it needs the observation of a wide signal percentage and the definition of a rich vocabulary.

Objective. In this work, a system able to recognize the user's current activity without a vocabulary of gestemes, and to accordingly adapt the manipulator's dynamic behavior is presented.

Methods and Material. An underlying stochastic model fits variations in the user's guidance forces and the resulting trajectories of the manipulator's end-effector with a set of Gaussian distribution. The high-level switching between these distributions is captured with hidden Markov models. The dynamic of the KUKA light-weight robot, a torque-controlled manipulator, is modified with respect to the classified activity using sigmoidal-shaped functions. The presented system is validated over a pool of 12 näive users in a scenario

Email address: federico.nessi@mail.polimi.it (Federico Nessi)
[1] Corresponding author.

that addresses surgical targeting tasks on soft tissue. The robot's assistance is adapted in order to obtain a stiff behavior during activities that require critical accuracy constraint, and higher compliance during wide movements. Both the ability to provide the correct classification at each moment (sample accuracy) and the capability of correctly identify the correct sequence of activity (sequence accuracy) were evaluated.

Results. The proposed classifier is fast and accurate in all the experiments conducted (80% sample accuracy after the observation of $\sim 450\ ms$ of signal). Moreover, the ability of recognize the correct sequence of activities, without unwanted transitions is guarenteed (sequence accuracy $\sim 90\%$ when computed far away from user desired transitions). Finally, the proposed activity-based adaptation of the robot's dynamic do not lead to a not smooth behavior (high smoothness, i.e. normalized jerk score < 0.01).

Conclusion. The provided system is able to dynamic assist the operator during cooperation in the presented scenario.

Keywords: activity recognition, cooperative robotics, context-awareness

1. Introduction

Cooperatively controlled robotic assistants are receiving greater acceptance in several application domains (e.g. industrial, medical) because of the advantages coming from the human agent involved in the control loop [1]. In hands-on robotic surgery, the surgeon moves tools fixed to the manipulator's end-effector by direct application of forces on the robot's links [2], achieving increased accuracy and safety during the operations, e.g. in retinal surgery [3] and orthopedic surgery [4]. Thus, hands-on controlled robots take advantage of the human decision making process and experience combining safety and intuitiveness with the enhancement strategies provided by the robot (e.g. hand tremor rejection or fatigue reduction) [5]. In assistive and rehabilitation robotics, for instance, the use of a manipulator proved to enhance post-trauma therapies [6]. The human user is often tightly coupled to the robotic device [6] and the use of the manipulator could provide active patient assistance in task-specific arm movement completion [7, 8] or guidance for a paretic arm during particular constrained movements [9]. In particular, the cooperative control approach in rehabilitation was proven to

enable patients to train in an active, variable and more natural way, with more physiological muscle activity [10].

To know *how* and *when* to provide the most appropriate level of assistance (e.g. in terms of adaptation of control strategy) is advisable in order to provide a more versatile robotic assistant and to get the best performances from the shared human-robot cooperation [11–13]. For example, during surgical brain cortex mapping procedures, the manipulator compliance can be adapted in order to damp the tool motion near the patient while maintaining highly compliant behavior elsewhere [14]. Conversely, in robot-aided gait rehabilitation, the robot assistance can be modified on the basis of the patient influence on the control, e.g. trigger leg movement whether a relevant muscle activity is detected [10].

In order to select the most suitable assistance at each moment it is necessary to infer the user's current activity/intention from raw input signals, online and in real-time [15]. In fact, effective human-robot interaction should avoid explicit UI mechanisms to change the assistant behavior [16]. Thus, a robot should be able to recognize the user's non-verbal cues [17], involving a degree of awareness of its surrounding [16].

Machine-learning algorithms are exploited in the field of assistive robotics to provide intuitive control of prosthesis (e.g. to predict switching between multiple functions of a powered artificial limb [6]) or to infer the user's intention of motion (e.g. to coordinate walking support exoskeletons for paraplegia patients [18]). They have also shown potential application in pre-surgical analysys, in order to classify different types of epilepsy in a fully automatic way [19].

In robotic surgery applications, intention-awareness has been addressed in order to distinguish whether a specific action, e.g. the violation of an active constraint [1], is intentionally performed, thus modifying the manipulator behavior when the action is intended, e.g. allowing the user violation [20]. Conversely, during surgical targeting tasks, activity recognition and online segmentation of the procedures into simple subtasks could allow to modify the modulation of the robot compliance with respect to the currently performed activity [12, 21].

A well-known approach to activity recognition is based on hidden Markov models (HMMs) [22]. HMMs are double-stochastic generative processes in which the observable output data is considered to be produced by a random variable taking values in a finite state space. Because of the very rich model's mathematical structure, HMMs are successfully applied to speech recogni-

tion [22], handwriting [23], gesture recognition [24, 25] as well as motion classification [26–28].

Based on the assumption that human motion actions can be split into a set of primitives (called *gestemes* [12]), high-level complex activities are considered to be produced by a temporal sequence of those primitives. The need to provide a complete set of gestemes have pointed out limitations in this approach, leading to an offline segmentation of performed tasks [12]. Furthermore, attempts to perform online classification were proved to be reliable (i.e. accuracy of $\sim 80\%$) only when observing a large percentage of the signal, i.e. more than 60% [29]. Finally, once a fast and robust recognition of the current surgeon's activity is provided, a robotic system able to adapt its behavior to the user's intention should include a strategy to smoothly switch among different control modalities [30].

In this work recognition of surgeon's activities during hands-on robotic surgery is aimed, to accordingly adapt the manipulator's behavior in order to augment the safety and intuitiveness of the cooperation. An online algorithm is presented, able to provide a robust and real-time recognition of surgeon's activities without the need to define gestemes. The underlying stochastic model [31, 32] fits the increments in the user's guidance forces and the resulting trajectory of the manipulator with the components of a Gaussian mixture model (GMM). The high-level switching between the different components is captured using a set of continuous HMMs, one for each activity. This is a more structured apprach with respect, e.g. to a black-box approach based on deep feedforward and recurrent neural networks to model processes with unknown states number and information length in time. Furthermore recurrent neural networks stability is not granted.

The provided classification is then exploited to trigger different dynamic behaviors of a torque-controlled manipulator. A strategy to switch among different behaviors is also presented, that modulates the robot stiffness and damping according to the user's activity.

The remainder of the paper is structured as follows. In Section 2, first the stochastic model used to describe user's actions is described, then the classification algorithm used to discriminate online among the activities is presented. In Section 3 the scenario and the selected modeled activities are described together with a strategy to smoothly modify the control of the assistant manipulator. Experimental evaluation of both the classifier and the adaptive robot control is presented in Section 4, encompassing a validation protocol over a pool of twelve näive users. Finally, results are presented in

Section 5 and discussed in Section 6.

2. Recognition of surgeon's activity

2.1. Activity model

Surgeon's activity during hands-on robotic surgery can be described by human cartesian driving forces ($\mathbf{f}$) and resulting manipulator's cartesian end-effector trajectory ($\mathbf{x}$), expressed as vectors of n-samples over time ($t = 1, \ldots, n$), i.e. $\mathbf{f} = (\mathbf{f}^1, \ldots, \mathbf{f}^n)$ and $\mathbf{x} = (\mathbf{x}^1, \ldots, \mathbf{x}^n)$. Both $\mathbf{f}$ and $\mathbf{x}$ are dependent on the current activity, thus they can be combined in a sequence of n 6-dimensional vectors $\mathbf{d}^t$ [31], i.e.

$$\mathbf{d} = (\mathbf{d}^1, \ldots, \mathbf{d}^n) = \begin{pmatrix} \mathbf{x} \\ \mathbf{f} \end{pmatrix}^T \tag{1}$$

that describes the user's action on the manipulator during cooperation.

The vector $\mathbf{d}$ can be expressed by a sequence of increments with respect to the current user's action, [32], i.e.

$$\mathbf{d}^t = \mathbf{d}^{t-1} + \Delta\mathbf{d}^t. \tag{2}$$

Vector $\Delta\mathbf{d}^t$ represents the increment at time t and is modeled as

$$\Delta\mathbf{d}^t = \mathbf{T}_{z_t} + \sqrt{\mathbf{C}_{z_t}} \cdot \mathbf{w}^t \tag{3}$$

thus being the emission of a 6-dimensional Gaussian distribution (*low-level* model) labeled $z_t \in \{1, \ldots, M\}$ with $\mathbf{T}_{z_t}$ mean and $\mathbf{C}_{z_t}$ covariance. Vector $\mathbf{w}^t$ is the sample of a zero-mean and identity covariance Gaussian random vector. Equation 2 is a switched-dynamical system of type $\frac{d}{dt}q^t = f_{z_t}(t)$, [33], with $f_{z_t} \equiv \Delta\mathbf{d}^t$ and Δt considered unitary, thus it is fully characterized by M Gaussian distributions defined by the $\mathbf{T} = (\mathbf{T}_1, \ldots, \mathbf{T}_M)$ and $\mathbf{C} = (\mathbf{C}_1, \ldots, \mathbf{C}_M)$ matrices, one at the time producing the current increment $\Delta\mathbf{d}^t$.

At higher-level, the temporal sequence of low-level models $\mathbf{z} = (z_1, \ldots, z_n)$ is considered as the sample of a Markov chain (*high-level* model) in which each state represents one of the M *low-level* models. Thus, each activity $a \in \{1, \ldots, A\}$ is characterized by one M-states continuous HMM $\lambda_a = \{\pi_a, \mathbf{B}_a, \mathbf{T}, \mathbf{C}\}$ in which the state emission probability density parameters are in fact the $\mathbf{T}$ and $\mathbf{C}$ matrices, π_a represents the prior probability of each

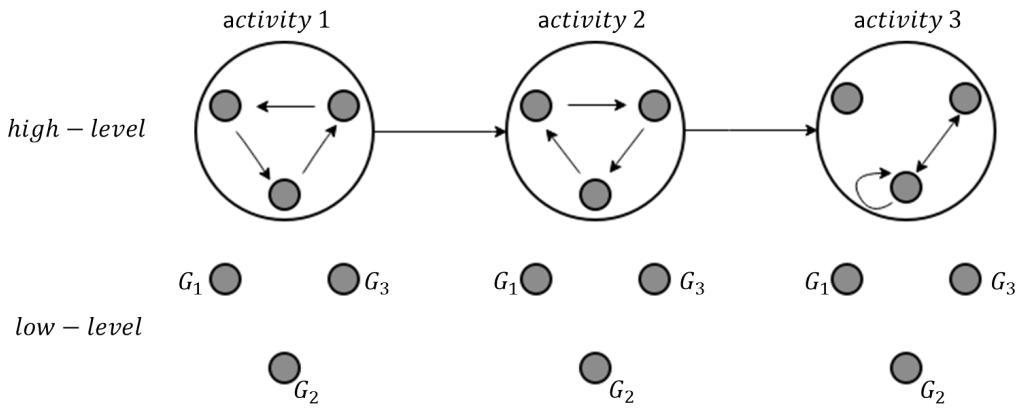

Figure 1: Representation of the two-level stochastic model that describes the user's activities. A set of Gaussian distributions that fit the increments in the user's guidance are identified in the *low-level*. The different switching pattern among the *low-level* distributions is characteristic of each activity and represent its *high-level* model. Moreover, a sequence of activities is represented.

state of the HMM and $\mathbf{B}_a$ is the transition matrix that characterize the *low-level* models switching for the activity a, i.e. $p(\mathbf{z}|a) = p(\mathbf{z}|\mathbf{B}_a, \pi_a)$.

This two-level hierarchical stochastic model is described in Figure 1.

2.2. Model training

The model is trained using a set of labeled data (i.e. it is known *a priori* which activity generated the data) produced by all the activities $a \in \{1, \ldots, A\}$ to be modeled.

In the first step, the parameters $\mathbf{T}$ and $\mathbf{C}$ describing the *low-level* models are estimated. The hypothesis made in this phase is that all the activities share the same models, that is the $\mathbf{T}$ and $\mathbf{C}$ matrices describe the state emission probability densities of all the HMMs [31, 32]. In such a way, the M *low-level* models can be considered as the components of an M-mixtures GMM with parameters $\boldsymbol{\theta} = \{\widehat{\mathbf{T}}, \widehat{\mathbf{C}}\}$. Thus the algorithm presented in [34] was used to estimate the parameters of $\boldsymbol{\theta}$ from increments computed directly over the complete set of training data, i.e.

$$\Delta\mathbf{d}^t = \mathbf{d}^t - \mathbf{d}^{t-1}. \tag{4}$$

The algorithm also selects the optimal number M of components for the GMM (i.e. the number of *low-level* models) in an unsupervised way, anni-

hilating redundant components starting from an initialization with an overestimated number of Gaussians.

In the second phase, the *high-level* models of the activities are obtained training one continuous HMM for each activity a. Since the emission probability density parameters are known from the *low-level* models estimate, the transition matrix $\mathbf{B}_a$ is simply obtained using a modified version of the Baum-Welch algorithm in which the emission parameters of each state z are imposed to $\widehat{\mathbf{T}}_z$ and $\widehat{\mathbf{C}}_z$ [29, 31, 32].

2.3. Classification algorithm

In order to provide online recognition of performed activities, the classifier must deal with incomplete set of data. For each new lot of data $\mathbf{d}^* \subset \mathbf{d}$ the class-conditional likelihood, i.e. $p(\mathbf{d}^*|\boldsymbol{\theta}, \mathbf{B}_a)$, is computed for all the trained HMMs using a forward-backward recursion [35], as reported in Appendix. The current activity is then selected as

$$\widehat{a} = \underset{a}{\operatorname{argmax}} \Big(p(\mathbf{d}^*|\boldsymbol{\theta}, \mathbf{B}_a) \Big) \tag{5}$$

The classification algorithm is implemented in C/C++ language inside the Robot Operating System[2] (ROS). Every iteration, the class-conditional likelihood of each activity is computed over data buffered in an array with first in first out (FIFO) logic [31]. This algorithm uses a fixed windowing (FW) strategy for the data buffer, i.e. the length of the buffered data is chosen *a priori* and kept fixed.

Determining the size of the data buffer is non-trivial: classification performed over arrays with fixed size leads to a trade-off between reliable classification during continuous activities (i.e. wide size of the buffer) and fast response to sudden transitions (i.e. short size of the buffer) [31, 36]. To address this problem, an adaptive windowing (AW) algorithm was implemented, that automatically increases the size of the buffer when the new available samples statistically reflects the already present ones, and shrinks it when a change is detected.

Because of the double-stochastic nature of HMMs, the proposed algorithm addresses window adaptation using a double-sided buffer. Both incoming data from the manipulator and class-conditional likelihood signals

[2] www.ros.org, Accessed: 23 May 2016

Algorithm 1 Adaptive windowing classification

procedure
  initialize S_d, S_{ll}
  declare $S_{ll}[1, \ldots, A][\ldots]$
  declare $ll[1, \ldots, A]$
loop:
  read $\mathbf{d}^*$
  $S_d \leftarrow S_d \cup \{\mathbf{d}^*\}$
  procedure CLASSIFICATION
    for each $a \in \{1, \ldots, A\}$ **do**
      $ll[a] \leftarrow p(S_d|\boldsymbol{\theta}, \mathbf{B}_a, \pi_a)$
    endfor
    $S_{ll} \leftarrow S_{ll} \cup \{ll\}$
    $\widehat{a} \leftarrow \underset{a}{\text{argmax}}(ll)$
  procedure WINDOW ADAPTATION
    for each $a \in \{1, \ldots, A\}$ **do**
      $S_B \leftarrow S_{ll}[a][1], \ldots, S_{ll}[a][1 - \rho N]$
      $S_A \leftarrow S_{ll}[a][1 - \rho N + 1], \ldots, S_{ll}[a][N]$
      $\mu_A \leftarrow mean(S_A)$, $\mu_B \leftarrow mean(S_B)$
      $\sigma_B \leftarrow std(S_B)$
      if $|\mu_A - \mu_B| > \epsilon \cdot \sigma_B$ **then**
        $S_d \leftarrow \mathbf{d}^*$
        break
      endif
    endfor
  provide $\widehat{a}$, *sizeof*(S_d) to system
  goto *loop*.

are buffered in stacks named S_d and S_{ll} respectively. The basic idea is that a drift detected in the likelihood signal of one activity should reflect a change in the probability that the current data are generated by the activity a. In order to detect such a drift, the class-conditional likelihood is computed every iteration and appended to S_{ll} buffer. The likelihood buffer is then split into two sub-windows S_A and S_B large ρN and $(1-\rho)N$ samples respectively, being N the size of S_{ll} and ρ the split fraction. Whenever the newest samples window S_A shows statistical difference with respect to the buffered S_B population of likelihood, S_d is flushed (but $\mathbf{d}^*$ is maintained). The test used to find statistical difference between the two sub-windows simply checks the following condition:

$$|\mu_A - \mu_B| > \epsilon \cdot \sigma_B \tag{6}$$

where μ_A and μ_B are the sample mean of S_A and S_B and σ_B is the standard deviation of S_B. This test is based on the fact that $\mu_A - \mu_B$ tends to a normal distribution for large window sizes [36]. The complete algorithm is reported in Algorithm 1.

3. Adaptation of robot's behavior

Hands-on guidance can be managed using a Cartesian torque-based impedance approach in order to exploit the high compliant dynamic behavior of flexible manipulators [37]. Desired Cartesian forces are thus computed as

$$\mathbf{f} = \mathbf{K}_P \cdot (\mathbf{x}_d - \mathbf{x}) + \mathbf{D}_P \cdot (\dot{\mathbf{x}}_d - \dot{\mathbf{x}}) \tag{7}$$

where $\mathbf{K}_P$ and $\mathbf{D}_P$ are the Cartesian stiffness and damping matrices of the manipulator arm (translational (P) component), $\mathbf{x}$ and $\mathbf{x}_d$ are the current and desired position of the control point at the robot end-effector respectively. A virtual fixture is applied to the rotational component in order to keep the end-effector orientation fixed. The dynamic of the robot is compensated with a feed-forward model-based torque controller, taking into account gravity and Coriolis-centrifugal terms.

Different stiffness and damping parameters allow to change the manipulator behavior, e.g. high resistance to user's guidance vs. great compliance, thus allowing different level of assistance.

In particular, in order to achieve high targeting accuracy, the damping characteristic of the manipulator can be modulated with respect to the distance from the desired target [38, 39]. Considering $\mathbf{K}_P$ and $\mathbf{D}_P$ isotropic

and diagonal, thus completely defined with one scalar parameter K_P and D_P, the space variable (SV) damping criterion [14] computes the D_P value (K_P is nil) as a function of the distance between the current position of the control point and the known position of the surgical target ($\mathbf{x}_T$), i.e.

$$D_P(\left\|\mathbf{x}-\mathbf{x}_T\right\|) = \underline{\mathrm{D}}_P + (\bar{D}_P - \underline{\mathrm{D}}_P)\frac{1}{1+e^{\beta\|\mathbf{x}-\mathbf{x}_T\|-m}} \tag{8}$$

where $\|\|$ denotes the Euclidean norm operator, $\underline{\mathrm{D}}_P$ and $\bar{D}_P$ are the lower and upper boundaries of the damping parameter, m is the spatial threshold that defines the radius of the isotropic sphere around the target in which the damping is steepest and β defines the damping change rate in the sigmoid function.

3.1. Scenario and robot state machine

We identifies four main activities performed by the surgeon, during which different levels of assistance are preferable:

- **approaching**: when approaching the patient's brain; the robot dynamic is modulated with the SV criterion based on the distance from the specific target of interest (null stiffness $K_P = 0\ N/m$, varying damping $D_P = 0 \div 30\ Ns/m$), in order to increase its damping behavior (thus applying more resistance to the user's interactive force) where high precision is required for the instrument's positioning.

- **leaving**: when leaving the patient's brain after the approaching is concluded; the robot is required to be as compliant as possible in order not to resist to the user's guidance, thus it is controlled in gravity compensation (null stiffness $K_P = 0\ N/m$, null damping $D_P = 0\ Ns/m$).

- **wandering**: when the manipulator is moved in the surgical area without a specific target, e.g. for the initial positioning or for the repositioning due to encumbrance issue; the robot is required to be very compliant in order to minimize the execution time and the surgeon's fatigue, thus it is controlled in gravity compensation (null stiffness $K_P = 0\ N/m$, null damping $D_P = 0\ Ns/m$).

- **idle**: when the manipulator is in a resting configuration; the robot is required to be stiff enough to maintain the configuration in case of unwanted interaction with the surgeon, thus it is controlled with fixed

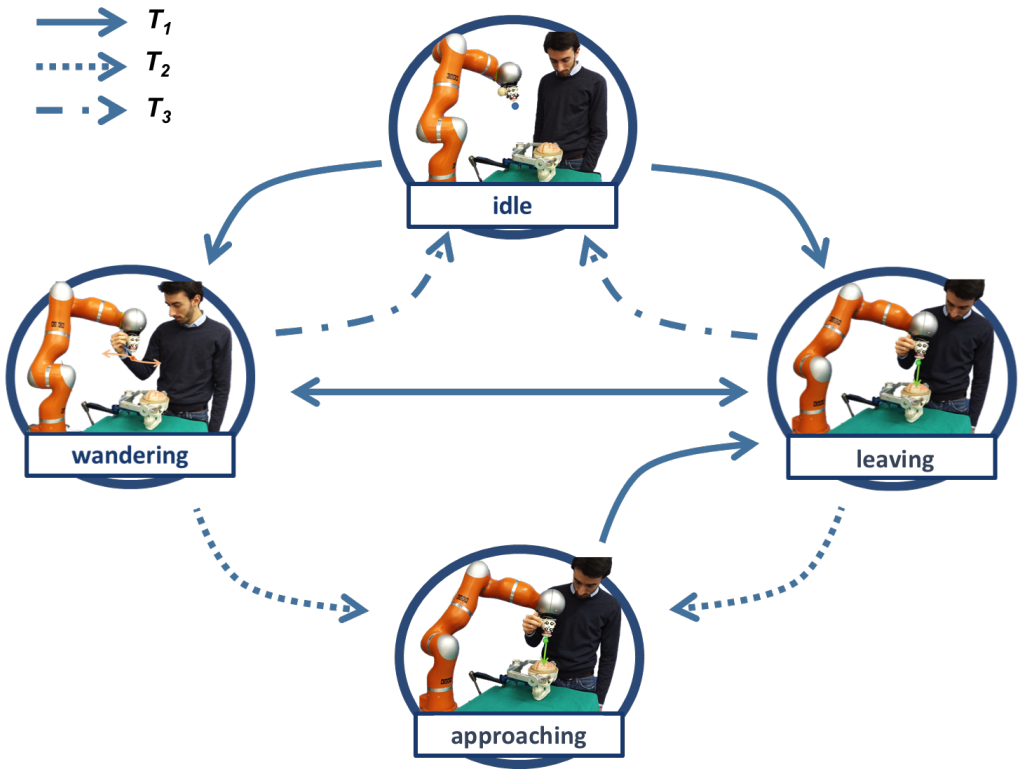

Figure 2: Scheme of the finite state machine to adapt the robot dynamic behavior during targeting tasks on soft tissues. The four states of the FSM represent the four activities performed by the surgeon identified for the brain cortex mapping procedure, i.e. *idle, wandering, approaching* and *leaving*. The three different transitions (T_{1-3}) between states are also shown.

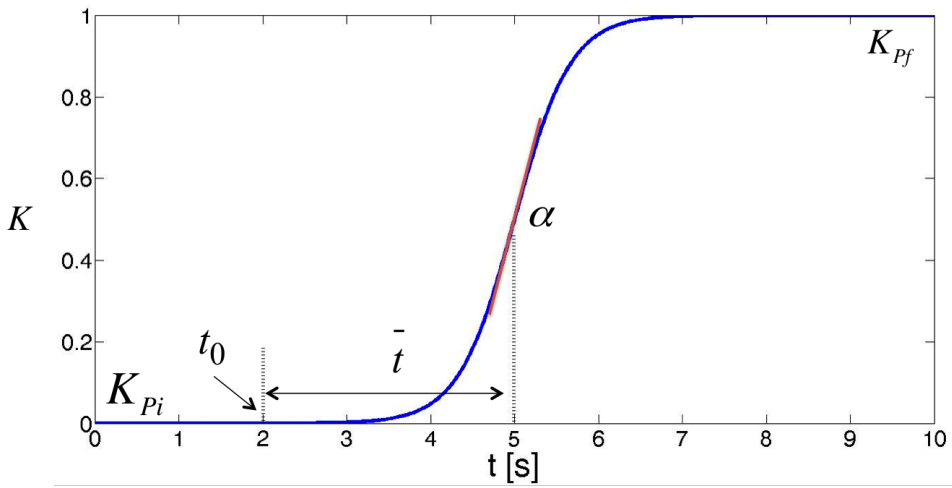

Figure 3: Sigmoid function used to vary the manipulator dynamics over time. P_i and P_f represent the initial (current state/activity) and final (desired state/activity) values of the dynamic characteristics respectively, t_0 is the time in which the transition is triggered, $\bar{t}$ is the time interval from t_0 when the modulation reaches its middle value (sigmoidal flex) and α represents the variation speed (sigmoid steepness)

dynamic parameters (stiffness $K_P = 100\ N/m$ and damping $K_D = 30\ Ns/m$).

The online recognition of the current activity performed by the surgeon (cfr. Paragraph 2.3) triggers the transition between the different states, which allows for the smooth modulation of the robotic dynamic parameters in a safe and controlled way during guidance and robot's motion. In particular, configurable time-varying sigmoidal functions were implemented to vary the stiffness ($K_P(t)$) and damping ($D_P(t)$) characteristics of the manipulator, i.e.

$$K_P(t) = K_{P_i} + \frac{(K_{P_f} - K_{P_i})}{1 + e^{-\alpha(|t-t_0|-\bar{t})}} \tag{9}$$

$$D_P(t) = D_{P_i} + \frac{(D_{P_f} - D_{P_i})}{1 + e^{-\alpha(|t-t_0|-\bar{t})}} \tag{10}$$

where i and f subscribes represents the initial (current state/activity) and final (desired state/activity) values of the dynamic characteristics respectively, t_0 is the time in which the transition is triggered, $\bar{t}$ is the time interval

from t_0 when the modulation reaches its middle value (sigmoidal flex) and α represents the variation speed (sigmoid steepness). The sigmoidal-shaped function is represented in Figure 3

The finite state machine (FSM) implemented to manage the system control behavior is shown in Figure 2. It has to be noted that the FSM is not fully-connected; in particular, the transitions from *approaching* to *idle* or *wandering* are not allowed for safety reasons: user's small correction around the target at the end of the *approaching* phase would be recognized as random movements, i.e. would trigger an unwanted *wandering* behavior. Following this approach, only small movements are allowed to the user in a region near to the POIs, i.e. at the and of the *approaching* phase. At the same time, some degree of motion of the manipulator should be recognized in the *wandering* or *leaving* activities before switching into the *approaching* phase to robustly recognize the intention of the user to move linearly towards a target.

Three different transitions among the four activities were implemented to smoothly switch between with different degrees of criticality due to contemporary presence of the human guidance during the controlled dynamic modulation, i.e.

- T_1: transitions towards the *idle* activity; the damping is modulated from the current value ($D_{P_i} = 0\ Ns/m$) to the maximum value ($D_{P_f} = 30\ Ns/m$) and the stiffness is modulated from the minumum value ($K_{P_i} = 0\ N/m$) to the maximum value ($D_{P_f} = 100\ N/m$).

- T_2: transitions towards *wandering* and *leaving*, the damping is modulated from the current value to the minimum value ($D_{P_f} = 0\ Ns/m$).

- T_3: transitions towards the *approaching* activity; the damping is modulated from the minimum value ($D_{P_i} = 0\ Ns/m$) to the current SV value ($D_{P_f} = D_P \cdot |\mathbf{x} - \mathbf{x}_T|$), computed online based on the actual position of the control point acquired during the transition trigger. The target $\mathbf{x}_T$ is previously acquired and considered known.

The supervisor FSM was implemented in C/C++ language inside the ROS and ORoCoS[3] frameworks.

[3]www.orocos.org, Accessed: 25 May 2016

4. Experimental evaluation

The chosen application scenario is soft tissue targeting during surgical intervention. A custom made surgical probe was attached to a light-weight robot 4+ (LWR4+, Kuka, Augsburg), a 7 degrees-of-freedom (DoFs) manipulator. The robot is guided through a 3D printed spherical handle placed at the end-effector. The guidance forces ($\mathbf{f}$) applied at the end-effector are estimated from the external torques measured by the manipulator's joint torque sensors. The resulting trajectory of the end-effector ($\mathbf{x}$) is obtained from the joint encoders. The control schema presented in reference [38] set the damping coefficient of the robot as a nonlinear function of the distance between the robot and the target, which was pre-operatively planned and intra-operatively registered using the robotic manipulator. Once the robot gets close to the target, only accurate and small movements are expected, so the damping is set to high. During the presented experimental trials, this control was deployed in a real-time environment with a control loop frequency of 200 Hz, though a Xenomai[4] patched Linux kernel. The experimental setup is represented in Figure 4.

4.1. Classifier training

The model $\boldsymbol{\theta}$ and each HMM were obtained from a dataset of 30 trials for each activity $a \in \{idle, wandering, leaving, approaching\}$ ($A = 4$) performed by one expert user, recorded (200 Hz) and off-line processed.

4.2. Design of the experiments

Twelve näive users (age 22(2), height 174(11) cm) were asked to partecipate to the experiments. Every user was previously informed about the nature of the experimental session and gave his/her consent. During each trial, the users were standing and guiding the robot without elbow support. A *trial supervisor* assisted the experiments and labeled the correct user's activity at each time, which was orally communicated by each user, in order to provide ground-truth. The exact time of activity switching was provided by the supervisor by means of a keyboard input in response to the user's vocal command.

Each user performed 3 experiments:

[4] www.xenomai.org, Accessed: 25 May 2016

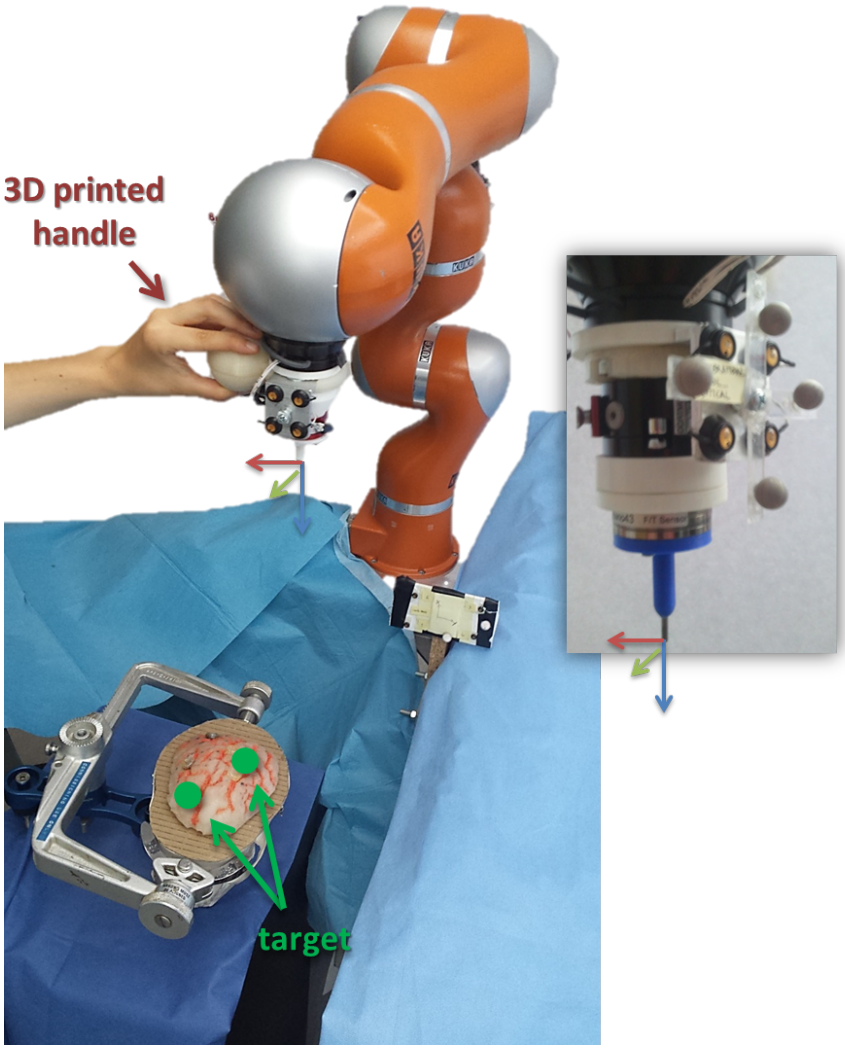

Figure 4: Experimental setup. The LWR4+ manipulator is guided through a 3D printed handle towards a set of targets on a brain phantom. User's Cartesian driving forces are recorded in the end-effector reference frame.

Experiment 1: assessment of the user-supervisor reaction time. Each user was instructed to give a sequence of 12 switching commands, randomly chosen between the possible activities, and to simultaneously press the space-bar on a keyboard. At the same time, the trial supervisor pressed the space-bar on another keyboard in response to the user's command. The time lag between the two button pressing was computed. This preliminary experiment was aimed at measuring the latency between the vocal command given by the user and the supervisor reaction when signaling the activity transition. This experiment did not involve the use of the manipulator.

Experiment 2: targeting task with fixed activity sequence. Each user was asked to perform 12 targeting tasks towards 12 arbitrary targets on a brain phantom following a predefined sequence of activities (i.e. *idle-wandering-approaching-idle-leaving-wandering-idle*). The proposed protocol includes 5 allowed and 1 denied transition in the robot control FSM for each trial. A vocal command was given by the user every time he/she switched between two activities, with a consequent keyboard input from the trial supervisor.

Experiment 3: targeting task with free activity sequence. The users was asked to perform 12 targeting tasks towards 12 arbitrary targets on the brain phantom, without a specific sequence of activities. They were anyway asked to signal activities switching by means of vocal commands. The ground-truth is provided by keyboard input from the supervisor in order to know the switching time, and by posterior off-line labeling of the switched activities.

During Experiment 2 and 3, the following algorithms were run on a dual-core Intel Xeon@2.66 GHz:

- **AW classifier**, as described in Paragraph 2.3. The algorithm was run at 20 Hz over a variable-size buffer with maximum size fixed at 150 samples (750 ms, $N = 150$). The window adaptation algorithm uses a split percentage of 5% ($\rho = 0.05$) with a confidence interval of 99% ($\epsilon = 3$). The output of the classifier was used to trigger the robot control FSM.

- **FW classifier**, the state-of-the-art classification algorithm that uses fixed-size buffer [31]. The algorithm was run at 20 Hz over a 50 samples long buffer in order to reach a trade-off between classification reliability

and fast responses. This algorithm was only used to validate the proposed AW classifier and thus its output do not trigger any transition in the robot control FSM.

The output of the two algorithms was recorded both sample-by-sample and in terms of activity transitions. In particular, each transition was labeled with respect to the ground-truth: every transition in the ground-truth signal was identified and considered against a time-window $[t_{transition}, \bar{t}_{latency}]$ around the transition time $t_{transition}$, being $\bar{t}_{latency}$ the maximum classification latency found in the current user's trial. Each classification is labeled as: *correct* if contained in the time-window and with final state equal to the ground-truth signal; *unseen* if no transition was found in the time-window. Every transition that do not follow these rules is labeled as *error*. The labeling procedure is represented in Figure 5

Moreover, the robot behavior was continuously adapted during Experiments 2 and 3 with respect to the current detected activity following what described in Section 3. Data regarding the current position of the surgical probe attached to the manipulator and user's guidance forces were recorded (200 Hz).

4.3. Classifier validation indexes

In order to evaluate the AW proposed classifier, the following performance indexes were computed for both AW and FW:

- *sample accuracy* (A_{sample}) [12], computed as the percentage of samples in the classification that share the same label with the ground-truth.
- *sequence accuracy* ($A_{sequence}$) [12], computed as

$$A_{sequence} = \frac{T_{correct}}{T_{classified} + T_{unseen}} \tag{11}$$

where $T_{correct}$ are the model transitions correctly identified, $T_{classified}$ are all the provided classification transitions and T_{unseen} are the missed transitions. This index directly evaluates the ability of the classifier to identify the correct sequence of transitions, without misclassification of missed transitions.

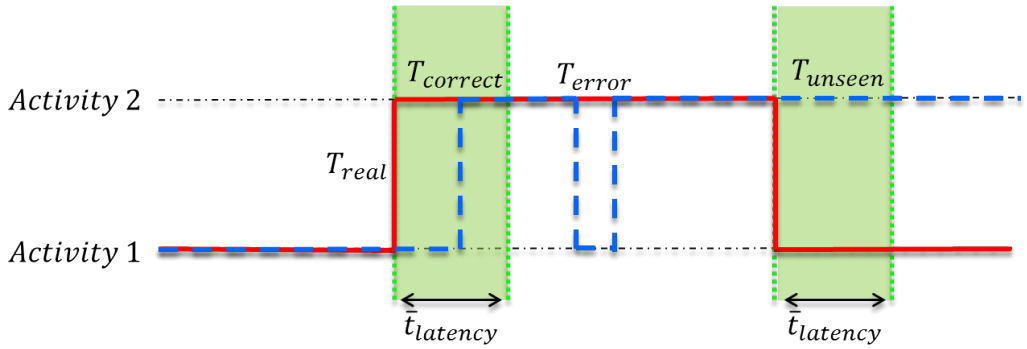

Figure 5: Labeling procedure for computing the $A_{sequence}$ index. The red solid line represents the ground-truth signal, the blue dashed line represents the classifier output while the green dotted lines show the time-window used in the labeling procedure. Each transition found outside that time-window is labeled as *error* and do not enter in the computation of the index.

- *latencies of classification* ($t_{latency}$), computed as the time between the ground-truth transition and the classifier reaction. The results are un-biased from the supervisor's reaction time computed during Experiment 1. An example of latencies is shown in Figure 6.

Because of the uncertainty that can happen during activity transitions (e.g. accelerations, decelerations, change of force direction, etc.) two more performance indexes are also computed, that do not consider errors during the transition window. These indexes assume that the classification during a transition cannot be considered reliable, thus validating the algorithm away from those time windows:

- *sample accuracy during continuous activities* (A^c_{sample}), computed like the sample accuracy but discarding the samples contained in the transition latency interval.

- *sequence accuracy during continuous activities* ($A^c_{sequence}$), computed like the sequence accuracy but not considering the transitions contained in the transition latency interval.

The median and interquartile range (25-75 %) values of each index were computed among users and results were compared using a Kruskal-Wallis test with Bonferroni correction ($p_{value} < 0.05$).

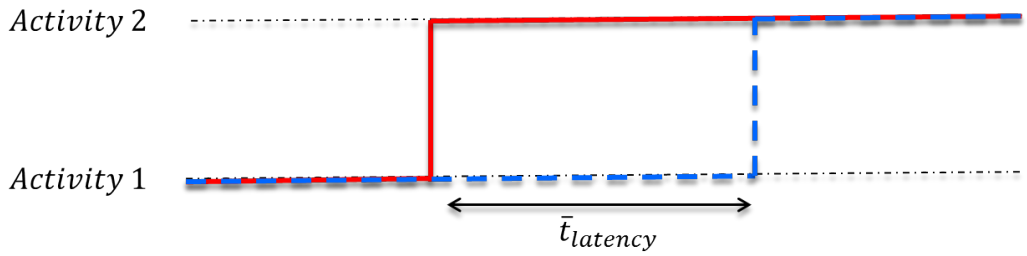

Figure 6: Example of transition latency $t_{latency}$. The red solid line represents the ground-truth activity switching while the blue dashed line represents the classifier response.

4.4. Adaptive control validation indexes

In previous publications we already demonstrated the stability and safety of the adaptive controller approach. In order to evaluate the smoothness of the robot dynamic behaviour changes, we computed the following index:

- *Normalized jerk score* (NJS) [40], computed as the root mean square integral jerk, i.e. third discrete derivative of the end-effector trajectory $\mathbf{x}$:

$$NJS = \sqrt{\frac{1}{2}\sum_{t=t_0}^{2\bar{t}} \left\| \left(\frac{d^3\mathbf{x}}{dt^3} \right)^i \right\|^2 \frac{T^5}{L^2}} \tag{12}$$

  where T and L are normalized factors, respectively equal to the trajectory duration and the trajectory length. L is computed as the discrete integral of the averaged (10 samples) position of the end-effector trajectory $\mathbf{x}$.

The NJS index of the guided robotic motion during the different allowed transitions was compared to the trajectory smoothness during each specific activity, except for the *idle* activity in which no motion of the manipulator occurs. The median and quartiles (25%, 75%) values of the NJS index were computed among users.

5. Results

5.1. Model training

The algorithm used to analyze the data describing the different activities fitted the increments with a mixture of 24-components Guassians (M =

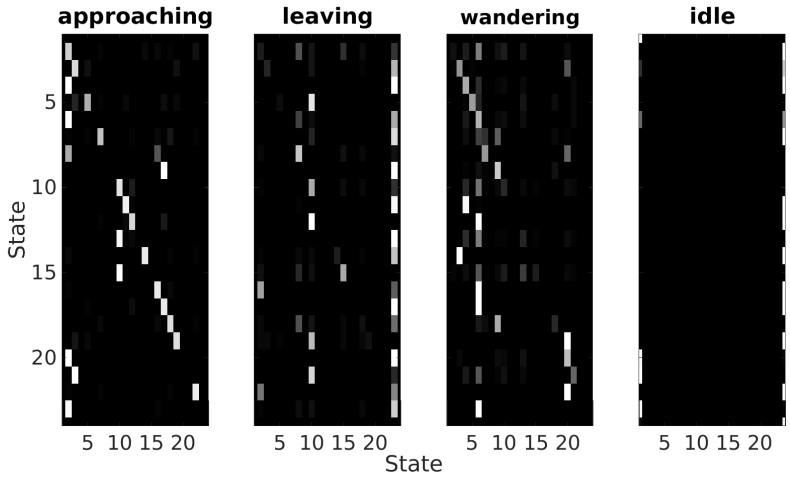

Figure 7: Results of the model fitting for the four identified activities. The transition matrix $\mathbf{B}_a$ of each trained HMM is represented in Figure, where the intensity of each block represents the transition probability between two states (white means probability equal to 1.0, black means probability equal to 0.0)

24). The transition matrices describing the different HMMs are reported in Figure 7. The *idle* activity is the emission of a zero-increment Gaussian distribution. The *approaching* matrix is similar to a diagonal matrix, i.e. the linear motion occurring when approaching the tissue can be assumed to be produced by a single Gaussian. Conversely, both the *leaving* and *wandering* activity matrices are more sparse, due to the random nature of the two movements.

5.2. Classifier Validation

Experiment 1. The assessment in the delay between the vocal command of the user and the keyboard reaction of the supervisor showed median values $\sim 300\ ms$ for each user.

Experiments 2 and 3. The proposed classifiers performed in a similar way (no statistically significant differences, Figure 8) to the state-of-the-art FW algorithm in terms of A_{sample}, thus guaranteeing a high overall accuracy ($\sim 80\%$) during the complete set of trials. The AW algorithm performed slightly better in terms of median value ($\sim 82\%$ vs. $\sim 79\%$ for experiment 2 and $\sim 78\%$ vs. $\sim 70\%$ for experiment 3). Moreover, the AW algorithm performed better in terms of $A_{sequence}$ in both Experiments ($\sim 60\%$ vs. $\sim 20\%$, statistically

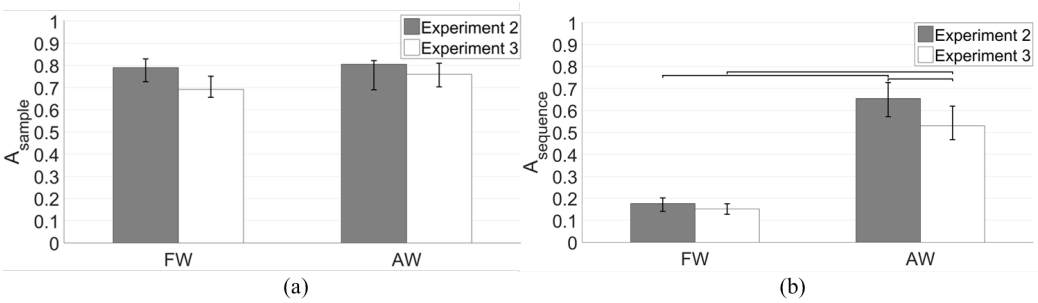

Figure 8: Results of A_{sample} (a) and $A_{sequence}$ (b) for both AW and FW algorithms. The proposed AW performed better than the state-of-the-art algorithm in terms of both indexes. Results are shown in terms of median and interquartile range (IQR). Horizontal lines represent statistically significant differences.

significant difference), thus strongly increasing the reliability of the identified sequence. AW has different performances during experiment 2 and 3 ($\sim 70\%$ vs. $\sim 60\%$ median values, statistically significant difference).

In terms of latencies ($t_{latency}$, Figure 9), the proposed algorithm performed worse than the state-of-the-art FW ($\sim 450\ ms$ vs. $\sim 250\ ms$, statistically significant difference) during experiment 2. Conversely, during experiment 3, the proposed AW algorithm have similar performances ($\sim 420\ ms$ vs. $\sim 450\ ms$, no statistically significant difference). Moreover, the AW algorithm strongly reduce the variability of the $t_{latency}$ index (inter-quartile range of $\sim 300\ ms$ vs. $\sim 800\ ms$).

In terms of A^c_{sample} index (Figure 10a), thus not considering the transition windows during the compute, the AW algorithm performed better than the state-of-the-art FW algorithm in all the Experiments (statistically significant difference). In particular, the AW algorithm guaranteed $\sim 90\%$ and $\sim 80\%$ accuracy during experiment 2 and 3 respectively (no statistically significant difference), while the accuracy provided by the FW algorithm was $\sim 80\%$ and $\sim 73\%$ during experiment 2 and 3 respectively (no statistically significant difference). Moreover, the FW algorithm shows similar results compared to the A_{sample} index (Figure 8a, median values of $\sim 80\%$ and $\sim 75\%$ for Experiments 2 and 3 respectively in both indexes) whereas the AW algorithm shows increased performance.

With respect to the $A^c_{sequence}$ index, both the AW and FW algorithm show increased performance (with respect to the $A_{sequence}$ index). In particular, AW performed better in both experiments ($\sim 90\%$ vs. $\sim 65\%$ and $\sim 80\%$

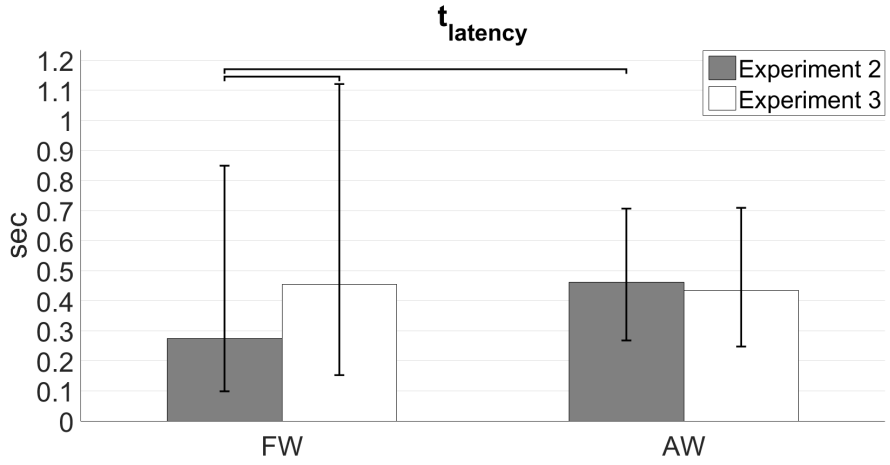

Figure 9: Results of $t_{latency}$ for both AW and FW algorithms. The proposed AW performed similarly to the state-of-the-art algorithm. Results are shown in terms of median and interquartile range (IQR). Horizontal lines represent statistically significant differences.

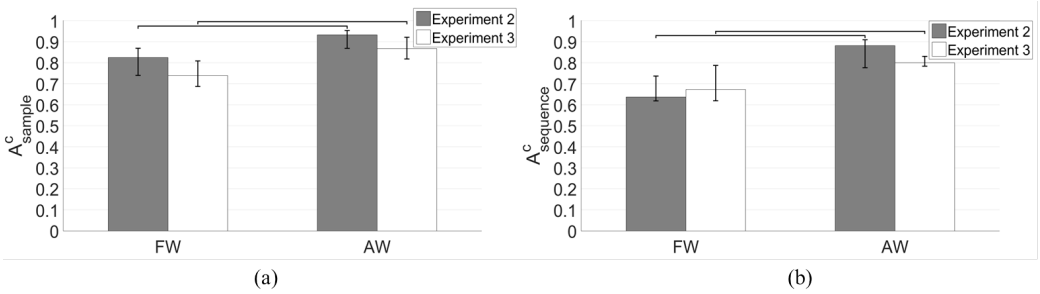

Figure 10: Results of A^c_{sample} (a), $A^c_{sequence}$ (b) for both AW and FW algorithms. The proposed AW performed better than the state-of-the-art algorithm in terms of both indexes. Results are shown in terms of median and interquartile range (IQR). Horizontal lines represent statistically significant differences.

vs. $\sim 67\%$ median values for Experiments 2 and 3 respectively, statistically significant differences).

These results are summarized in Table 2.

In Table 2 the activity switching probabilities recorded during the experiment 3 (targeting tasks without a predefined sequence of activities) are reported. The values are based on the user's feedback regarding the current performed activities, thus not involve errors in classification. It can be noted

Table 1: Summary of the classification accuracy of both AW and FW during experiment 2 and 3. Results are reported in terms of median values.

	A_{sample}	$A_{sequence}$	A^c_{sample}	$A^c_{sequence}$
	Experiment 2			
FW	∼0.79	∼0.20	∼0.80	∼0.65
AW	∼0.82	∼0.60	∼0.90	∼0.90
	Experiment 3			
FW	∼0.70	∼0.19	∼0.73	∼0.67
AW	∼0.78	∼0.50	∼0.80	∼0.80

Table 2: Transition matrix during experiment 3. In Table the percentage of switching between activities performed by the users during the free sequence trial is represented. Only transitions are considered, thus the diagonal is imposed to 0.

		to			
		approaching	**leaving**	**wandering**	**idle**
from	**approaching**	0	0.0123	0.1790	0.8086
	leaving	0.1894	0	0.5985	0.2121
	wandering	0.5421	0.0579	0	0.4000
	idle	0.1441	0.3517	0.5042	0

how the *idle* and *approaching* activities have a well-defined target activity (*approaching* switching to *idle* in 80% of the cases, *idle* switching to *wandering* in 50% of the transitions), whereas the *wandering* activity has almost equally distributed transition.

5.3. Adaptive control validation

Motion smoothness during transitions and states is reported in Figure 11, while in Table 3 the quantitative results are summarized. The NJS indexes of the states (median value below 0.03) resulted comparable to the one of the considered transitions (median value below 0.04) except for the transitions from/to *idle* activity. In the latter, although the smoothness is reduced (median NJS value around 0.1 and 0.05 when switching to *idle* from *wandering* and *leaving* respectively) due to the switch between damping and high stiffness control, the order of magnitude of the NJS index is the same, thus still limited. Also, the high NJS variability in the transition *wandering-idle* (IQR above 0.08) indicates how its smoothness is affected by the user's execution

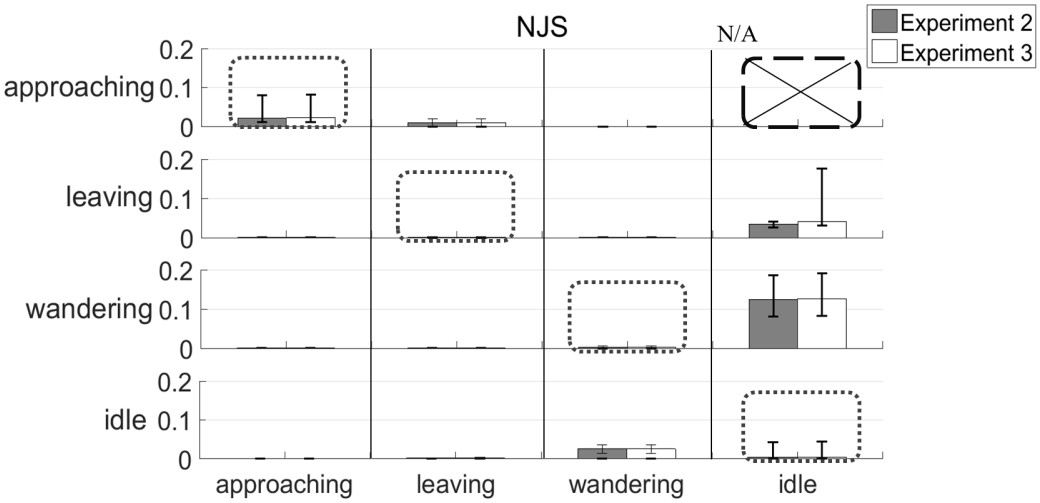

Figure 11: Results of the NJS index computed on the allowed transitions from the initial activities (rows) to the final activity (columns) among users. Values highlighted with dashed lines represents the NJS computed during the execution of a continuous activity, while the not allowed transitions are shown with a crossed symbol. Quantitative results are summarized in Table 3

during start/stop procedures.

The NJS index computed in the approaching state confirms that the SV variable damping controller allows for smooth guidance comparable to the smoothness performance of the gravity compensations controller, i.e. highest transparent/compliant controller.

6. Discussions

This work presents a system for robotic adaptive assistance which could be applied in several Human-Robot cooperative scenarios. In particular is focused on the assistance to surgeons during soft tissue targeting tasks. Because of the need to online recognize intended activities, the system must be highly accurate (e.g. 80% accuracy) while maintaining a low latency response to not degrade the interactivity of the user experience (the trade-off of which is reported to be around 700 ms [41]). Classification based on sequences of primitives is not suitable: the classification accuracy hits 80% only when observing more than 60% of the signal [29], which implies a latency possibly in the order of some seconds. Moreover, these results were obtained training

Table 3: Median values of NJS index result computed on the allowed transitions from the initial activities (rows) to the final activity (columns) among users. Values not reported are considered negligible or related to transition not allowed (n.a.).

		to			
		approaching	**leaving**	**wandering**	**idle**
from	**approaching**	∼0.02 - ∼0.02	∼0.005 - ∼0.005	-	n.a.
	leaving	-	-	-	∼0.03 -∼0.04
	wandering	-	-	-	∼0.05 -∼0.05
	idle	-	-	∼0.2 -∼0.2	-

the primitives on the same person that would have then performed the task [29]. Our classifier manages to accurately classify activities with a latency of $\sim 450\ ms$ in median, and to be operator independent. The difference between the proposed approach and the literature make an actual experimental comparison not possible, since the other reported approaches does not satisfy the given requirements for the interaction.

The presented algorithm addresses gesteme-free activity classification (without the need to define a vocabulary of gestemes). Surgeon's activities are modeled with a set of Gaussian distributions (*low-level* models) and the switching between these models is represented with different HMMs (*high-level* models). The classification is performed on data buffered in array with variable size in order to avoid typical trade-off between reliability and fast response. The behavior of the torque-controlled manipulator is adapted accordingly to the current detected activity. The presented system can be applied in real scenarios with a clear utility: improve the operators performance and increase the patient safety and surgery quality.

The system was tested over a pool of twelve näive users. Each user was asked to perform targeting gestures over a brain phantom first following a predefined sequence of activities and then following a self-paced sequence. During the experiments, a trial supervisor labeled each performed activity with respect to the user's vocal feedback to provide ground-truth, while the time of activity switching was known by means of keyboard input from the supervisor. The time lag between the user's vocal command and the supervisor reaction was previously assessed for each user and resulted comparable to the human reaction time ($\sim 300\ ms$, [42]). The proposed protocol is different from other works with similar scope, e.g. in [12] the user's feedback was given using a pedal, where it was stated that the usage of an external

device during the cooperation could be distracting for the user. The vocal command used allows each user to concentrate on the task, but can involve delays between the command uttering and the actual switch in the activity that are not measurable.

The proposed gesteme-free classifier was validated against a state-of-the-art classification algorithm that uses fixed size buffer (FW). The performance of the two algorithms were assessed in terms of ability of providing the correct classification at each moment (*sample accuracy*). The A_{sample} index is strongly affected by the algorithms reaction time: the higher the latency, the more samples in the classification will not correspond to the provided ground-truth during transitions. In fact, an algorithm able to fast react to activity transitions without guaranteeing reliability (i.e. with misclassification during continuous activities) will obtain a high A_{sample} result. For this reason, also the reliability of that classification was evaluated, i.e. ability to provide the correct sequence of activities without misclassification (*sequence accuracy*). Moreover, the reaction to both the algorithms to sudden activity transitions was measured (*latency*).

Experimental results showed that the proposed AW algorithm is accurate ($\sim$ 80% *sample accuracy*) and reliable (*sequence accuracy* over 60%). FW showed faster response during experiment 2 (median values of $\sim 250\ ms$ vs. $\sim 450\ ms$) but worse reliability. This is related to the small buffer used by FW (50 samples, fixed windowing), that allows fast response but is sensitive to small changes in the guidance data. Conversely, during experiment 3 the AW algorithm performed similarly to FW in terms of latencies (median values $\sim 450\ ms$), obtaining an almost 10% increase in the A_{sample} performance. Since the method is used to assist the surgeon and not to substitute him or her, the obtained recognition ratio mean than 80 times over 100 the assistance is provided, while in the remaining 20 the system works almost as a non assisting device, i.e. a standard hands on robot. It must be noted that this is the result obtained so far, but further improvements are possible by. e.g. allowing the user to trigger a different behaviour before the current transition is completed.

Both the algorithms are affected by activity transitions: accelerations and decelerations during activity switching lead to misclassification during the transition window that eventually converge to the correct classification when the user's guidance stabilizes. This is confirmed when the transition windows are not considered: AW and FW showed increased performances. In particular, AW performed better leading to a $\sim$ 90% accuracy in both

Experiments 2 and 3 and both indexes. These results are comparable with state-of-the-art offline classifiers based on *gestemes*, e.g. [12] where peg-in-hole tasks performed with the JHU Steady Hand were segmented with *sample accuracy* of the same order of magnitude.

The classification errors during transitions do not lead to a not smooth behavior of the manipulator, as demonstrated by results of the NJS index. In fact, only transition from/to *idle* activity showed a reduced smoothness due to the users stopping/accelerating strategy. In particular, the *wandering-approaching*, *approaching-leaving* and *wandering-leaving* transitions showed NJS value negligible. These results are endorsed by the users sequence choices during experiment 3 (switching probabilities represented in Table 2), where the highlighted transitions resulted to be the most addressed. Moreover, these probabilities could be used in order to provide a different way of triggering transitions: if an identified transition has high-probability it can be immediately executed, waiting for more information in case of transitions with low probability.

The main limitation of this work was the number of modeled activities. Our surgical model is based of four surgery activities (a state machine with idle, leaving, wandering and approaching statuses): in order to express the complexity of a surgery, the number of recognized activity should increase. We expect that the training phase would be longer, but still the different activities would be correctly classified. As said, this paper is a proof of concept for online gesture classification and robot behaviour adaptation. A more complex surgical scenario or other non-surgical scenarios could possibly be modeled with a wider set of activities, thus leading to a more in-depth segmentation of the surgical procedure. Other critical aspects not addressed were the duration of the time-varying transition in the robot dynamic behavior. In particular, some users pointed out that fast switching between two activities before a transition was completed lead to a not intuitive behavior of the manipulator, e.g. increasing of the stiffness while trying to trigger a *wandering* compliant behavior. Moreover, the application of the SV damping criterion [14] during *approaching* was based on a target known *a priori*. The implementation of a target prediction algorithm could overcome this limitation, applying the SV maximum damping around the identified user's current target. It would be also possible to define some task oriented behaviour of the robot. This can be achieved adding more states on the finite states machine. Also, the implemented control does not allow the user to exploit also rotations of the end-effector of the manipulator: in our surgical

scenario, the robot is assisting the surgeon in achieving precise alignment on pre-operatively defined target points, therefore the orientation is not affecting the results accuracy. Indeed (e.g. in case of straight electrodes insertion) the orientation also could be included and this would be straightforward in our current implementation. Finally, the inclusion in the system of a force-feedback enhancement criterion can further increase the surgeon confidence in the task [43].

7. Conclusions

This paper presented the first results of a system to provide a gesteme-free adaptive assistance to surgeon's during hands-on controlled cooperation in surgical robotics. The system was proven to correctly classify the current user's action (classification accuracy of $\sim 80\%$ after $\sim 450\ ms$), adapting the manipulator behavior accordingly to a set of identified activities. Validation encompassed targeting gesture trials in a simulated surgical scenario. Future works may include implementation of target prediction in order to better apply space variable damping criteria, as well as a validation phase involving expert and/or novice surgeons. Additionally, trajectory identification during approaching phase may be accomplished in order to modulate the stiffness ellipsoid of the manipulator, for example addressing human-like behavior.

Appendix A. Model's likelihood

Given a sequence of observation $\mathbf{O} = \{O_1, \ldots, O_T\}$ and a known sequence of states $\mathbf{Q} = \{q_1, \ldots, q_T\}$ of a Model λ, the probability of observing the sequence $\mathbf{O}$ produced by the states sequence $\mathbf{Q}$ is

$$P(\mathbf{O}|\mathbf{Q}, \lambda) = \prod_{t=1}^{T} P(O_t|q_t, \lambda) \tag{A.1}$$

Considering the state sequence $\mathbf{Q}$ as unknown, the probability of observing simultaneously $\mathbf{O}$ and $\mathbf{Q}$ is

$$P(\mathbf{O}, \mathbf{Q}|\lambda) = P(\mathbf{O}|\mathbf{Q}, \lambda) \cdot P(\mathbf{Q}|\lambda) \tag{A.2}$$

and thus the posterior probability of $\mathbf{O}$ given the model (i.e. the model likelihood) can be computed as the probability of the observation sequence to be produced by all the possible sequence of states, i.e.

$$P(\mathbf{O}|\lambda) = \sum_{\forall \mathbf{Q}} P(\mathbf{O}|\mathbf{Q}, \lambda) \cdot P(\mathbf{Q}|\lambda) \tag{A.3}$$

Considering a N-states continuous HMM $\lambda = \{\mathbf{A}, \mathbf{B}, \pi\}$, the elements describing the model have the following significance: $\mathbf{A}[N \times N]$ is the transition matrix containing the probability of switching between the model's state; $\pi[N \times 1]$ is the prior probability of being in one of the N states; $\mathbf{B}[N \times 1]$ is the set of probability distributions that characterize the emission of the n-th model state, i.e.

$$P(O_i|q_j, \lambda) = b_j(O_i) \tag{A.4}$$
$$P(q_i|q_j, \lambda) = a_{ij} \tag{A.5}$$
$$P(q_i|\lambda) = \pi_i \tag{A.6}$$

The probability of observing O_t and be in state $q_t = S_i$ at time t, i.e. $\alpha_t(i) = P(\{O_1, \ldots, O_t\}, q_t = S_i)$, can thus be computed by recursively solving

$$\alpha_{t=1}(i) = \pi_i \cdot b_i(O_1) \tag{A.7}$$
$$\alpha_{t+1}(j) = b_j(O_{t+1}) \cdot \sum_{i=1}^{N} \alpha_t(i) \cdot a_{ij} \tag{A.8}$$

Equations A.7-A.8 are the so-called *foward recursion* and its termination has the following meaning

$$\alpha_T(i) = P(\{O_1, \ldots, O_T\}, q_T = S_i|\lambda) \tag{A.9}$$

thus leading to simply compute the observation sequence likelihood as

$$P(\mathbf{O}|\lambda) = \sum_{i=1}^{N} \alpha_T(i) \tag{A.10}$$

Acknowledgements

This work was funded by the FP7 ACTIVE project (FP7-ICT-2009-6-270460).

References

[1] B. Davies, M. Jakopec, S. J. Harris, A. Barrett, A. Evangelidis, P. Gomes, et al., Active-constraint robotics for surgery, Proceedings of the IEEE 94 (9) (2006) 1696–1704.

[2] B. L. Davies, S. J. Harris, F. R. y Baena, P. Gomes, M. Jakopec, Hands-on robotic surgery: is this the future?, Medical Imaging and Augmented Reality (2004) 27–37.

[3] A. Üneri, M. Balicki, J. Handa, P. Gehlbach, R. H. Taylor, I. Iordachita, et al., New steady-hand eye robot with micro-force sensing for vitreo-retinal surgery, in: Biomedical Robotics and Biomechatronics (BioRob), 2010 3rd IEEE RAS and EMBS International Conference on, IEEE, IEEE, Tokyo, Japan, 2010, pp. 814–819.

[4] A. D. Pearle, D. Kendoff, V. Stueber, V. Musahl, J. A. Repicci, Perioperative management of unicompartmental knee arthroplasty using the mako robotic arm system (makoplasty), American Journal of Orthopedics 38 (2) (2009) 16–19.

[5] R. MacLachlan, B. C. Becker, J. Cuevas Tabares, G. W. Podnar, L. Lobes, C. N. Riviere, et al., Micron: an actively stabilized handheld tool for microsurgery, Robotics, IEEE Transactions on 28 (1) (2012) 195–212.

[6] P. M. Pilarski, M. R. Dawson, T. Degris, J. P. Carey, R. S. Sutton, Dynamic switching and real-time machine learning for improved human control of assistive biomedical robots, in: Biomedical Robotics and Biomechatronics (BioRob), 2012 4th IEEE RAS & EMBS International Conference on, IEEE, IEEE, Rome, Italy, 2012, pp. 296–302.

[7] D. J. Reinkensmeyer, L. E. Kahn, M. Averbuch, A. McKenna-Cole, B. D. Schmit, W. Z. Rymer, Understanding and treating arm movement impairment after chronic brain injury: progress with the arm guide, Journal of rehabilitation research and development 37 (6) (2000) 653–662.

[8] A. Pedrocchi, S. Ferrante, E. Ambrosini, M. Gandolla, C. Casellato, T. Schauer, et al., Mundus project: Multimodal neuroprosthesis for daily upper limb support, J Neuroeng Rehabil 10 (2013) 66.

[9] P. S. Lum, C. G. Burgar, P. C. Shor, M. Majmundar, M. Van der Loos, Robot-assisted movement training compared with conventional therapy techniques for the rehabilitation of upper-limb motor function after stroke, Archives of physical medicine and rehabilitation 83 (7) (2002) 952–959.

[10] A. Duschau-Wicke, J. Von Zitzewitz, A. Caprez, L. Lünenburger, R. Riener, Path control: a method for patient-cooperative robot-aided gait rehabilitation, Neural Systems and Rehabilitation Engineering, IEEE Transactions on 18 (1) (2010) 38–48.

[11] M. Li, A. M. Okamura, Recognition of operator motions for real-time assistance using virtual fixtures, in: Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2003. HAPTICS 2003. Proceedings. 11th Symposium on, IEEE, IEEE, Los Angeles, CA, 2003, pp. 125–131.

[12] C. S. Hundtofte, G. D. Hager, A. M. Okamura, Building a task language for segmentation and recognition of user input to cooperative manipulation systems, in: Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2002. HAPTICS 2002. Proceedings. 10th Symposium on, IEEE, 2002, pp. 225–230.

[13] A. Lüdtke, D. Javaux, F. Tango, R. Heers, K. Benglere, C. Ronflé-Nadaud, Designing dynamic distributed cooperative human-machine systems, Work 41 (Supplement 1) (2012) 4250–4257.

[14] E. Beretta, E. De Momi, F. R. y Baena, G. Ferrigno, Adaptive hands-on control for reaching and targeting tasks in surgery, International Journal of Advanced Robotic Systems 12 (2015) 50.

[15] K. Hauser, Recognition, prediction, and planning for assisted teleoperation of freeform tasks, Autonomous Robots 35 (4) (2013) 241–254.

[16] S. A. Green, M. Billinghurst, X. Chen, G. Chase, Human-robot collaboration: A literature review and augmented reality approach in design.

[17] O. C. Schrempf, U. D. Hanebeck, A. J. Schmid, H. Worn, A novel approach to proactive human-robot cooperation, in: Robot and Human Interactive Communication, 2005. ROMAN 2005. IEEE International Workshop on, IEEE, IEEE, 2005, pp. 555–560.

[18] K. Suzuki, G. Mito, H. Kawamoto, Y. Hasegawa, Y. Sankai, Intention-based walking support for paraplegia patients with robot suit hal, Advanced Robotics 21 (12) (2007) 1441–1469.

[19] Y. Kassahun, R. Perrone, E. De Momi, E. Berghöfer, L. Tassi, M. P. Canevini, et al., Automatic classification of epilepsy types using ontology-based and genetics-based machine learning, Artificial intelligence in medicine 61 (2) (2014) 79–88.

[20] J. G. Petersen, F. Rodriguez Baena, A dynamic active constraints approach for hands-on robotic surgery, in: Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on, IEEE, IEEE, 2013, pp. 1966–1971.

[21] W. Yu, R. Alqasemi, R. Dubey, N. Pernalete, Telemanipulation assistance based on motion intention recognition, in: Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on, IEEE, IEEE, Barcelona, Spain, 2005, pp. 1121–1126.

[22] L. R. Rabiner, A tutorial on hidden markov models and selected applications in speech recognition, Proceedings of the IEEE 77 (2) (1989) 257–286.

[23] A. Brakensiek, A. Kosmala, D. Willett, W. Wang, G. Rigoll, Performance evaluation of a new hybrid modeling technique for handwriting recognition using identical on-line and off-line data, in: Document Analysis and Recognition, 1999. ICDAR'99. Proceedings of the Fifth International Conference on, IEEE, IEEE, Bangalore, India, 1999, pp. 446–449.

[24] K. Liu, C. Chen, R. Jafari, N. Kehtarnavaz, Multi-hmm classification for hand gesture recognition using two differing modality sensors, in: Circuits and Systems Conference (DCAS), 2014 IEEE Dallas, IEEE, IEEE, Dallas, USA, 2014, pp. 1–4.

[25] R.-H. Liang, M. Ouhyoung, A real-time continuous gesture recognition system for sign language, in: Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on, IEEE, IEEE, 1998, pp. 558–567.

[26] D. Aarno, D. Kragic, Layered hmm for motion intention recognition, in: 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, Bejing, 2006, pp. 5130–5135. doi:10.1109/IROS.2006.282606.

[27] Z. Wang, A. Peer, M. Buss, An hmm approach to realistic haptic human-robot interaction, in: EuroHaptics conference, 2009 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics 2009. Third Joint, IEEE, Salt Lake City, (Utah, US), 2009, pp. 374–379. doi:10.1109/WHC.2009.4810835.

[28] E. Gribovskaya, A. Kheddar, A. Billard, Motion learning and adaptive impedance for robot control during physical interaction with humans, in: Robotics and Automation (ICRA), 2011 IEEE International Conference on, IEEE, Shangai, 2011, pp. 4326–4332. doi:10.1109/ICRA.2011.5980070.

[29] D. Aarno, D. Kragic, Motion intention recognition in robot assisted applications, Robotics and Autonomous Systems 56 (8) (2008) 692–705.

[30] M. D. Comparetti, E. Beretta, M. Kunze, E. De Momi, J. Raczkowsky, G. Ferrigno, Event-based device-behavior switching in surgical human-robot interaction, in: Robotics and Automation (ICRA), 2014 IEEE International Conference on, IEEE, IEEE, Hong Kong, China, 2014, pp. 1877–1882.

[31] F. Nessi, E. Beretta, G. Ferrigno, E. De Momi, Recogntition of user's activity for adaptive cooperative assistance in robotic surgery, in: Engineering in Medicine and Biology Society (EMBC), 2015 IEEE/EMBS International Conference on, IEEE/EMBS, IEEE, Milan, Italy, 2015.

[32] J. C. Nascimento, M. A. Figueiredo, J. S. Marques, Trajectory classification using switched dynamical hidden markov models, Image Processing, IEEE Transactions on 19 (5) (2010) 1338–1348.

[33] Z. Sun, S. S. Ge, Stability theory of switched dynamical systems, Springer Science & Business Media, London, 2011.

[34] M. A. Figueiredo, A. K. Jain, Unsupervised learning of finite mixture models, Pattern Analysis and Machine Intelligence, IEEE Transactions on 24 (3) (2002) 381–396.

[35] L. E. Baum, J. A. Eagon, et al., An inequality with applications to statistical estimation for probabilistic functions of markov processes and to a model for ecology, Bull. Amer. Math. Soc 73 (3) (1967) 360–363.

[36] A. Bifet, R. Gavalda, Learning from time-changing data with adaptive windowing., in: SDM SIAM International Conference on Data Mining, Vol. 7, SIAM, 2007.

[37] A. Albu-Schaffer, C. Ott, G. Hirzinger, A passivity based cartesian impedance controller for flexible joint robots – part II: Full state feedback, impedance design and experiments, in: Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on, Vol. 3, IEEE, Barcelona, Spain, 2004, pp. 2666–2672.

[38] E. Beretta, F. Nessi, G. Ferrigno, F. Di Meco, A. Perin, L. Bello, G. Casaceli, F. Raneri, A. De Benedictis, E. De Momi, Enhanced torque-based impedance control to assist brain targeting during open-skull neurosurgery: a feasibility study, The International Journal of Medical Robotics and Computer Assisted Surgery (2015) 50.

[39] E. Beretta, G. Ferrigno, E. De Momi, Nonlinear force feedback enhancement for cooperative robotic neurosurgery enforces virtual boundaries on cortex surface, Journal of Medical Robotics Research 01 (02) (2016) 1650001. doi:10.1142/S2424905X1650001X.

[40] J.-J. Chang, T.-I. Wu, W.-L. Wu, F.-C. Su, Kinematical measure for spastic reaching in children with cerebral palsy, Clinical Biomechanics 20 (4) (2005) 381–388.

[41] C. Ellis, S. Z. Masood, M. F. Tappen, J. J. LaViola, R. Sukthankar, Exploring the trade-off between accuracy and observational latency in action recognition, International Journal of Computer Vision 101 (3) (2013) 420–436. doi:10.1007/s11263-012-0550-7.
URL http://dx.doi.org/10.1007/s11263-012-0550-7

[42] J. Shelton, G. P. Kumar, Comparison between auditory and visual simple reaction times, Neuroscience & Medicine 1 (01) (2010) 30.

[43] E. Beretta, F. Nessi, G. Ferrigno, E. D. Momi, Force feedback enhancement for soft tissue interaction tasks in cooperative robotic

surgery, in: Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on, IEEE, Hamburg, 2015, pp. 209–215. doi:10.1109/IROS.2015.7353376.