Null Space Redundancy Learning for a Flexible Surgical Robot

Danilo Bruno*, Sylvain Calinon*† and Darwin G. Caldwell*
*Department of Advanced Robotics - Istituto Italiano di Tecnologia (IIT) - Via Morego, 30 - 16163 Genova - Italy
Email: name.surname@iit.it
†Idiap Research Institute, CH-1920 Martigny, Switzerland

***Abstract*— A new challenge for surgical robotics is placed in the use of flexible manipulators, to perform procedures that are impossible for currently available rigid robots. Since the surgeon only controls the end-effector of the manipulator, new control strategies need to be developed to correctly move its flexible body without damaging the surrounding environment. This paper shows how a positional controller for a new surgical robot (STIFF-FLOP) can be learnt from the demonstrations given by an expert user. The proposed algorithm exploits the variability of the task to comply with the constraints only when needed, by implementing a minimal intervention principle control strategy. The results are applied to scenarios involving movements inside a constrained environment and disturbance rejection.**

I. INTRODUCTION

In minimally invasive surgery, tools go through narrow openings and manipulate soft organs to perform surgical tasks. There are limitations to current robot-assisted surgical systems due to the rigidity of robot tools. The aim of the STIFF-FLOP European project is to develop a soft robotic arm to perform surgical tasks by actively controlling the selected body parts of the robot [1], [2], [3]. The flexibility of the robot allows the surgeon to move within organs to reach remote areas of the body and perform challenging procedures in laparoscopy.

The surgeon controls the end-effector during the surgical task, leaving the motion of the whole arm to the control and learning modules. The latter should drive the body of the robot along the trajectory followed by the surgeon, without applying pressure to or damaging the internal organs of the patients. The learning algorithm will have to work in the null space of the surgical manipulator, to avoid interfering with the surgeon and exploiting redundancy in an optimal way [4], [5].

Indeed, the shape of the robot cannot always be the same as the trajectory. Because of the limitations of its geometry, the robot cannot fit the trajectory with the whole body all the time. One way to do this is to create an algorithm that keeps the robot close to the tracked trajectory where the constraints need to be satisfied more strictly and exploits the places where more freedom is available to fulfill the task precisely.

We explore the use of learning from demonstrations techniques to program the correct behaviour of the surgical robot, having as input the expert demonstrations provided by the surgeons. These demonstrations will have to show the possible trajectories that are needed to perform the task, thus allowing the algorithm to extract information on the task variability and constraints. Surgeons often require to set some constraint along the motion, where critical situation might occur. Within the current framework, the constraints define viapoints as low variability points in the demonstrations, where the robot will be forced to pass.

The variability of the demonstrations is encoded in a compact statistical model by using Gaussian Mixture Models (GMM) whence the information can be efficiently extracted at reproduction time by means of Gaussian Mixture Regression (GMR) [6]. Alternative approaches to encode the variability of the data can be used without affecting the overall procedure, such as Gaussian Processes [7] or Hidden Markov Models (HMM).

Since the timing of the task during the reproduction phase is given by the surgeon, a new application of GMR is proposed. The inputs are based on curvilinear distances and a time-invariant mechanism to control the robot in a surgical scenario is obtained. This results into a novel application of GMR, which has been used in the literature for motion generation of autonomouse systems [8], [9], for coupling dynamical systems of dynamic movement primitives [10], [11] or to extract trajectories from time-indexed demonstrations [12].

This approach also differs from usual algorithms using Operational Space Control from demonstrations [13]: in our case no hierarchical priority is defined inside the null space, but the importance of each null space operation depends on the variability of the task and changes online. This approach can be eventually coupled with standard techniques: if a prior hierarchy is defined in null-space, the proposed approach can be used to control the remaining available DOFs.

These ideas also find a correspondence within the framework of the minimal intervention principle [14], stating that the actions on the system should be limited to those really affecting the completion of the task. In our case, forcing the robot to always stay near the given trajectory could limit the possible configurations and constrain the task execution too much.

The technique described in this paper can be used both to control the motion of the robot and to determine the directions or the areas that are more relevant for disturbance rejection. We performed experiments showing both situations. Since the robot is still under development, an

This work was partially supported by the STIFF-FLOP European project under contract FP7-ICT-287728. Website:www.stiff-flop.eu

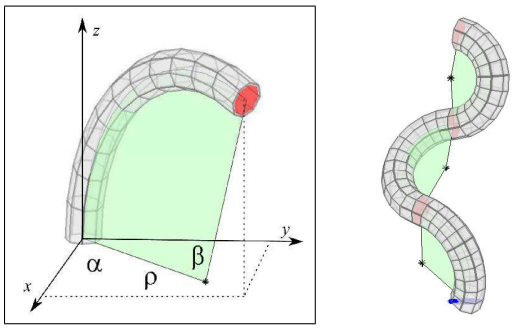

Fig. 1. The MatLab Inverse Kinematics simulator of the STIFF-FLOP robot. Left: a single module modelled as a constant curvature module. Right: the full robot, obtained putting 3 sections together.

inverse kinematics simulator in Matlab is used to run the experiments, see also [15].

II. THE STIFF-FLOP ROBOT

The first prototype of the robot, currently under development, is composed of 3 cylindrical sections (links). Each link of the robot consists of a soft cylinder with three chambers disposed concentrically around the axis, where air is inflated to bend the link in the desired orientation. A central chamber filled with hard grain-shaped particles is used to stiffen the link at a desired orientation by air suction [1], [2].

Different models were developed in the literature to describe the kinematics of such flexible modules [16], In most cases, our robot can be modeled as a constant curvature section of a circle. In its local frame, the rest position (no chamber is inflated) corresponds to the module aligned along the vertical axis $\hat{\boldsymbol{z}}$, with a rest length L_0. The present prototype of the single module is $0.05m$ long in the rest position and has a diameter of $0.04m$. When totally inflated, it can elongate by 80%. Moreover, each link can bend at approximatively 180^0.

The position $\boldsymbol{q}_i$ of the tip of the i-th module can be written as a function of the angle α_i, arc length β_i and curvature radius ρ_i as follows:

$$\boldsymbol{q}_i = \rho_i[(1-\cos(\beta_i))\cos(\alpha_i), (1-\cos(\beta_i))\sin(\alpha_i), \sin(\beta_i)] \tag{1}$$

Both variables $\boldsymbol{q}_i$ and $\rho_i, \alpha_i, \beta_i$ can be used to describe the kinematics of the module. The Cartesian coordinates of any point along the single module can be written as a function of its fractional position ξ

$$\boldsymbol{f}_{\rho_i,\alpha_i,\beta_i}(\xi) = \begin{bmatrix} \rho_i(1-\cos(\beta_i\xi))\cos(\alpha_i) \\ \rho_i(1-\cos(\beta_i\xi))\sin(\alpha_i) \\ \rho_i \sin(\beta_i\xi) \end{bmatrix}. \tag{2}$$

Here $\xi \in [0,1]$ corresponds to all possible points from the base of the module to the tip ($\xi = 0$ is the base).

From now on, we will use the Cartesian positions $\boldsymbol{q}_i$ of the tips of the modules in the rest frame of the base as internal variables. They will play the role of joint variables inside the following kinematical model.

The orientation of the tip frame only depends on its position, evaluated by rotating the tip frame to make $\hat{\boldsymbol{z}}$ tangent to the module at the tip, keeping the other axes rigidly displaced along the manipulator. The tip orientation of the i-th module in the $(i-1)$-th tip frame is respectively defined by a matrix $\boldsymbol{R}_{(i-1)i} = \boldsymbol{R}_{(i-1)i}(\boldsymbol{q}_i)$.

There are limitations on the possible configuration, since the inflation mechanism only allows a limited range of elongations of each chamber, depending on the bending of the module and on its orientation in space [15].

This setup allows an easy integration of multiple robot links, since any additional module can be thought as a constant curvature model applied on the previous. The position and orientation of the tip of the robot can be recursively evaluated for any possible number of links N. For 3 links this results into:

$$\boldsymbol{t} = \boldsymbol{q}_1 + \boldsymbol{R}_{01}\boldsymbol{q}_2 + \boldsymbol{R}_{01}\boldsymbol{R}_{12}\boldsymbol{q}_3, \quad \boldsymbol{R}_{03} = \boldsymbol{R}_{01}\boldsymbol{R}_{12}\boldsymbol{R}_{23}. \tag{3}$$

The whole orientation of the tip is not actually needed, since the rotation of the tip around the vertical axis will be performed during surgery by using the tool mounted on it. So, only the direction $\hat{\boldsymbol{z}}$ of the tip axis will be controlled, resulting into a 2-dimensional quaternion based orientation vector $\boldsymbol{\theta}$. The task parameters for the manipulator are the position $\boldsymbol{t}$ of the tip and its orientation $\boldsymbol{\theta}$, collected into a 5-dimensional task vector $\boldsymbol{w} = [\boldsymbol{t}, \boldsymbol{\theta}]^\top$.

The kinematics of the module is finally complemented by adding motion capabilities to the base of the manipulator (6 additional DOF, i.e. 3 Cartesian positions $\boldsymbol{q}_0$ corresponding to the translation of the base and 3 Euler angles $\boldsymbol{\eta}_0$ corresponding to all the possible rotations of the base). The rotation matrix corresponding to $\boldsymbol{\eta}_0$ will be denoted by $\boldsymbol{R}_0$.

So, the whole manipulator is endowed with a total number of $3N+6$ degrees of freedom, N being the number of links. The total joint coordinates will be denoted by

$$\hat{\boldsymbol{q}} = [\boldsymbol{\eta}_0, \boldsymbol{q}_0, \boldsymbol{q}_1, \ldots, \boldsymbol{q}_N]. \tag{4}$$

The position and orientation of the end effector are:

$$\boldsymbol{t} = \boldsymbol{q}_0 + \boldsymbol{R}_0 \left(\sum_{i=1}^{N} \boldsymbol{R}_{0,i-1}\boldsymbol{q}_i \right), \quad \boldsymbol{R}_{0N} = \boldsymbol{R}_0 \prod_{i=1}^{N} \boldsymbol{R}_{i-1,i} \tag{5}$$

The direct kinematics is represented by the function $\boldsymbol{w} = \boldsymbol{w}(\hat{\boldsymbol{q}})$; the inverse differential kinematics is considered, by evaluating the Jacobian $\boldsymbol{J}$ of the direct kinematics and using standard robotics techniques, with the internal variables replacing the role of joints, namely

$$\frac{d\boldsymbol{w}}{dt} = \frac{\partial \boldsymbol{w}}{\partial \hat{\boldsymbol{q}}} \frac{d\hat{\boldsymbol{q}}}{dt} = \boldsymbol{J}\frac{d\hat{\boldsymbol{q}}}{dt}, \tag{6}$$

where J is the Jacobian matrix. Given a starting position for the robot, corresponding to a choice of the internal parameters corresponding to $\hat{\boldsymbol{q}}(0)$ and task parameters $\boldsymbol{w}(0)$, the final configuration can be represented using the minimum norm solution as:

$$\hat{\boldsymbol{q}}(t) = \int_0^t \boldsymbol{J}^\dagger \frac{d\boldsymbol{w}}{dt} dt + \hat{\boldsymbol{q}}(0), \tag{7}$$

where $\frac{d\boldsymbol{w}}{dt}$ is the velocity of the movement in the task space and $\boldsymbol{J}^{\dagger}$ is the pseudoinverse of the Jacobian matrix. Since the above coordinates do not introduce singularities in the calculation of the Jacobian, this approach is already precise enough for motion generation. The knowledge of $\hat{\boldsymbol{q}}(t)$ allows us to know the configuration of each link of the manipulator with time and to actuate them using a low level positional control.

A 3 links manipulator will be endowed with 15 DOF: given the 5-dimensional task space, it has 10 DOF of redundancy that can be used to control the body, while the surgeon moves the tip to perform the surgical task. Learning will be performed in the null space, so that the body can be controlled without affecting the task kinematics.

The motion of any intermediate point is obtained by calculating its Jacobian and projecting the inverse differential kinematics on the null space. An internal coordinate $s \in [0, N]$ is assigned to the robot, specifying the rest position of all the points of the back bone of the manipulator ($s = 0$ corresponds to the base and $s = N$ to the end-effector). The Cartesian position can be then obtained by using eq.(2) in conjunction with eq.(5) as:

$$\boldsymbol{P}(s) = \boldsymbol{q}_0 + \boldsymbol{R}_0 \left[\sum_{i=0}^{\lfloor s \rfloor} \boldsymbol{R}_{0,i} \boldsymbol{q}_{i+1} + \boldsymbol{f}_{\rho_{\lfloor s \rfloor+1}, \alpha_{\lfloor s \rfloor+1}, \beta_{\lfloor s \rfloor+1}}(s - \lfloor s \rfloor) \right] \tag{8}$$

Here $\lfloor s \rfloor$ is the floor function applied to s. The intermediate Jacobian can be evaluated as

$$\hat{\boldsymbol{J}}(s) = \frac{\partial \boldsymbol{P}(s)}{\partial \hat{\boldsymbol{q}}}, \tag{9}$$

while the joint trajectory corresponding to a (null space) movement of the intermediate point $\boldsymbol{P}(s)$ is

$$\hat{\boldsymbol{q}}_0(t) = \int_0^t \left[\left(\boldsymbol{I} - \boldsymbol{J}^{\dagger} \boldsymbol{J} \right) \hat{\boldsymbol{J}}^{\dagger}(s) \frac{d\boldsymbol{P}}{dt} \right] dt + \hat{\boldsymbol{q}}_0(0), \tag{10}$$

where $\boldsymbol{J}$ is the Jacobian of the task space evaluated in eq.(6). In the following we will denote by $\boldsymbol{P}_i = \boldsymbol{P}(i), i = 1 \dots N$ the intermediate points corresponding to the conjunction between two modules.

III. PROBLEM SETTING

During the demonstration phase, the user provides the variability information of the task by demonstration. In this first phase, the variability is collected by demonstrating several possible paths that the constraints allow and are provided with different durations and timing distortions to be close to a real-world application. We will discuss in Section V how this approach can be generalized to be applied within a surgical context.

The demonstrations are encoded using a Gaussian Mixture Model (GMM), that is able to keep the variability information through covariance matrices. The GMM will encode the extended dataset

$$\boldsymbol{d} = [l, \boldsymbol{x}]^{\top} \sim \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{d} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{11}$$

where l denotes the curvilinear distance of the current point from the start of the trajectory and $\boldsymbol{x}$ is the Cartesian position along the trajectory. Moreover, $\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$ respectively represent the mixing coefficients, mean vectors and covariance matrices of K components GMM. As we shall see, the length information can be used to learn a desired elongation for each module.

In this paper, the needed information will be extracted from demonstrations by using Gaussian Mixture Regression (GMR). Each of the K Gaussians of the GMM encoding the trajectories are described by center and covariance

$$\boldsymbol{\mu}_k = \begin{bmatrix} \mu_k^l \\ \boldsymbol{\mu}_k^{\boldsymbol{x}} \end{bmatrix}, \quad \boldsymbol{\Sigma}_k = \begin{bmatrix} \Sigma_k^l & \boldsymbol{\Sigma}_k^{l\boldsymbol{x}} \\ \boldsymbol{\Sigma}_k^{\boldsymbol{x}l} & \boldsymbol{\Sigma}_k^{\boldsymbol{x}} \end{bmatrix} \tag{12}$$

and the conditional probabilities $\mathcal{P}(\boldsymbol{x}|l)$ and $\mathcal{P}(l|\boldsymbol{x})$ are modelled as Normal distributions i.e.

$$\boldsymbol{x}|l \sim \mathcal{N}(\boldsymbol{x} | \hat{\boldsymbol{\mu}}^{\boldsymbol{x}}, \hat{\boldsymbol{\Sigma}}^{\boldsymbol{x}}) \quad l|\boldsymbol{x} \sim \mathcal{N}(l | \hat{\boldsymbol{\mu}}^{l}, \hat{\boldsymbol{\Sigma}}^{l}) \tag{13}$$

In order to evaluate the mean and the covariance, let us denote by $_O$ the actual output variable index ($\boldsymbol{x}$ and l respectively), by $_I$ the conditioning index (l and $\boldsymbol{x}$ respectively) and by $\boldsymbol{u}^I$ the conditioning variable. We have that:

$$\hat{\boldsymbol{\mu}}^O = \sum_{i=1}^{K} h_i(\boldsymbol{u}^I) \left[\boldsymbol{\mu}_i^O + \boldsymbol{\Sigma}_i^{OI} \boldsymbol{\Sigma}_i^{I\,-1} (\boldsymbol{u}^I - \boldsymbol{\mu}_i^I) \right], \tag{14}$$

$$\hat{\boldsymbol{\Sigma}}^O = \sum_{i=1}^{K} h_i^2(\boldsymbol{u}^I) \left[\boldsymbol{\Sigma}_i^O - \boldsymbol{\Sigma}_i^{OI} \boldsymbol{\Sigma}_i^{I\,-1} \boldsymbol{\Sigma}_i^{IO} \right], \tag{15}$$

$$\text{where} \quad h_i(\boldsymbol{u}^I) = \frac{\pi_i \mathcal{N}(\boldsymbol{u}^I | \, \boldsymbol{\mu}_i^I, \boldsymbol{\Sigma}_i^I)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{u}^I | \, \boldsymbol{\mu}_k^I, \boldsymbol{\Sigma}_k^I)}. \tag{16}$$

First, the average trajectory $\hat{\boldsymbol{\mu}}^O$ is extracted from demonstrations by GMR; while the tip moves along this trajectory, the learning algorithm will infer the behaviour for the rest of the body of the STIFF-FLOP manipulator. This is done by determining the desired position of each junction point and the variability of demonstrations around this point, by using GMR with l as an input parameter, where l is evaluated assigning a desired distance from the other junction points. For instance, it can be chosen to correspond to the middle of the elongation range of the given module.

The output of the resulting algorithm is a positional controller for the intermediate points of the manipulator aimed at keeping them close to the mean trajectory within the error provided by the covariance information $\hat{\boldsymbol{\Sigma}}^O$. This Robot Body Behaviour Control (RBBC) is summarized in Algorithm 1 (See also the accompanying video[1]) .

The covariance information is taken into account by appropriately weighting the displacement vector moving the point $\boldsymbol{P}_i$ towards the desired position, that will be inversely proportional to the corresponding covariance.

The null space controller is implemented in terms of soft constraints: after each displacement is projected in the null space, the final command in joint space is evaluated

[1] http://programming-by-demonstration.org/icra14/

Input : GMM encoding the demonstrations $(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k)$ as in eq.(11)
Output: Displacement vector in the null space of STIFF-FLOP robot for all time steps
Initialize the desired length L of the modules
for *every time step* **do**
  for *each point* $\boldsymbol{P}_i$ *inside the body* **do**
    Evaluate the Jacobian of $\boldsymbol{P}_i$ by using eq.(8) with $s = i$
    Calculate the length of the trajectory inside the body by using GMR: $L_{in} = \hat{\boldsymbol{\mu}}^{\boldsymbol{P}_{tip}}$
    Evaluate the desired position of the points $\boldsymbol{P}_i$ by using GMR: $\boldsymbol{A}_i = \hat{\boldsymbol{\mu}}^{L_{in}-iL}$
    Evaluate the covariance $\tilde{\boldsymbol{\Sigma}}_i = \hat{\boldsymbol{\Sigma}}^{L_{in}-iL}$
    Calculate the importance factor

$$\alpha = \frac{\mathcal{N}(\boldsymbol{P}_i|\boldsymbol{A}_i, \tilde{\boldsymbol{\Sigma}}_i)}{\mathcal{N}(\boldsymbol{A}_i|\boldsymbol{A}_i, \tilde{\boldsymbol{\Sigma}}_i)} \in [0, 1] \qquad (17)$$

    Calculate the importance weighted displacement vector $\boldsymbol{V}_i = (1-\alpha)(\boldsymbol{A}_i - \boldsymbol{P}_i)$
    Calculate the null space velocity corresponding to the displacement vector as $\delta\boldsymbol{q}_i = (\boldsymbol{I} - \boldsymbol{J}_i^{\dagger}\boldsymbol{J}_i)\boldsymbol{J}_i^{\dagger}\boldsymbol{V}_i$
  Move the robot by making the average over the N_{inside} points inside the body at current time step

$$\delta\boldsymbol{q} = \frac{1}{N_{inside}} \sum_i \delta\boldsymbol{q}_i$$

Algorithm 1: Robot body behaviour control (RBBC).

by making the average of the joint space displacements corresponding to each $\boldsymbol{P}_i$ inside the body. In this way, none of the constraints is fulfilled exactly but the more relevant ones are weighted as having a higher importance and correspond to a bigger displacement. This behaviour is what differentiates the current approach from Operation Space Control [17] and is aimed at dynamically keeping a higher priority on displacements corresponding to a lower covariance. Obviously, if the task is structured in such a way that a null space command is known to have a higher priority, an Operational Space Control projection layer can be used before the RBBC algorithm is used.

Finally, we can observe that even though a desired elongation is set for each module as an input to the algorithm, the robot will use a slight range of elongations around the given one, because of the soft constraints policy.

The proposed approach affects the behaviour of the robot in two different ways. First of all, the macroscopic movements of the manipulator are always kept inside a safe area around the desired trajectory, that depends on the variability of the demonstrations at each point.

A second important aspect regards disturbance rejection. During the task, the presence of noise can move the manipulator away from the desired trajectory: in this sense, the RBBC algorithm will apply a variability dependence disturbance rejection policy.

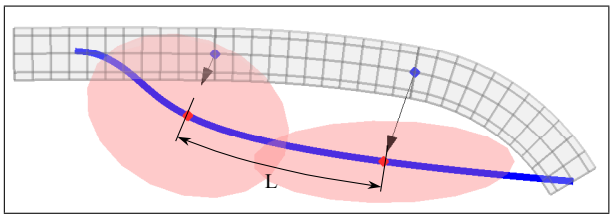

Fig. 2. Graphic explanation of Algorithm 1. The blue trajectory is the desired path for the body of the robot, while the demonstrated variability is represented by the red GMM. The red dots on the trajectory represent the learnt position of the attractors, based on the desired length of each module. The black arrows represent the displacements that are calculated in the null space of the robot.

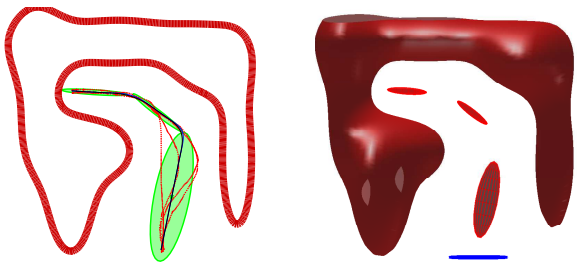

Fig. 3. Left: the demonstrations provided to the system on a plane; the variability depends on the possible movements inside the simulated scenario. Organs are represented by the red shape. The green ellipsoids represent the Gaussians of the GMM encoding the task. Right: the GMM resulting from the demonstrations inside the 3D environment; the blue line represents the tracking error for the trocar port.

IV. SIMULATION RESULTS

A virtual environment is defined in Matlab, with a representations of human organs. The organs are defined as level surfaces of a 3-dimensional GMM, i.e.

$$\mathcal{A} = \left\{ \boldsymbol{o} \middle| \sum_{k=1}^{K^{\text{organs}}} \pi_k^{organs} \mathcal{N}(\boldsymbol{o}|\boldsymbol{\mu}_k^{\text{organs}}, \boldsymbol{\Sigma}_k^{\text{organs}}) = const \right\}$$

The task is demonstrated in 2 dimensions, even though the environment and the robot are defined in the 3D environment: this is used to simplify the recordings of demonstrations and the visualization of the results. A series of paths is collected by controlling the tool with the mouse, representing some of the possible trajectories that can be used to perform the task. The collected demonstrations are shown in fig.3.

A certain amount of noise is added to the joints during the motion, to simulate the inaccuracy in the control of the robot. A null space controller is used to keep the robot inside the trocar port within a given tracking error during the task. The soft control policy is applied to the trocar port exactly as it is applied to other points along the manipulator: at each time step, the arm index of the point at the level of the trocatr port is calculated and a full displacement is applied to keep it in the center of the trocar port; this displacement is then averaged with other contributions from other points. Another

possible choice could be to define the trocar port with task space variables or to use Operational Space Control to put hard constraints on that point: yet, in current applications, surgeons report that they can apply some force on the trocar port to move the tools inside the body, resulting into a soft constraint request for the learning algorithm. For this reason, a desired tracking error is set for the trocar port and checked during the procedure.

As a first experiment, a trajectory passing between two organs and reaching a target behind one of them is proposed. This movement exemplifies one of the typical surgical movements that the robot is aimed to carry out, which is impossible for currently available rigid surgical robots. During the approach phase, the orientation of the tip of the robot is always kept tangent to the trajectory.

While the tip is precisely controlled by the surgeon, if the motion of the body of the robot is not corrected, it touches one of the organs with its body (Fig. 4d). When the RBBC algorithm is applied, the robot moves in-between the organs while performing the task (Fig. 4e-h).

At the beginning of the trajectory (Fig 4a,e), where there is high variability in the demonstrations, the free motion and the controlled one are identical, since no intervention is needed, even though the body of the robot is not exactly following the trajectory. As the lower variability part of the trajectory is reached, the body of the robot is kept nearer the desired path (Fig 4f,g), while the behaviour without corrections would be different (Fig 4b,c). Finally, the higher variability in the lower part of the trajectory allows the controller to steer the robot even more to follow the trajectory as precisely as possible in the final part by displacing the robot body in the region of high variability (Fig. 4h).

The RBBC algorithm thus allows the system to move the robot very precisely where this is needed and exploiting the variability of the motion where it is possible.

As an example of the usefulness of this minimal intervention approach, we show in Fig. 5 what happens if we try to always keep the robot as near as possible to the desired trajectory. As we can see on the right, the inverse kinematics without correction does not provide an appropriate compromise between the tracking error for the trocar port, the orientation of the end-effector and the position of the middle point along the trajectory. As a result the robot is going outside the tracking error for the trocar port and an unnecessarily high force is put onto the abdominal wall. Instead, once the variability is taken into account, the least important constaint is treated with lower priority and the robot fulfills the task successfully (Fig. 5-Right).

In a second experiment, the disturbance rejection effects are studied. The robot is moved along a trajectory with a different level of variability along it. The variability is low at the beginning and at the end of the movement. A Gaussian noise with $\sigma = 0.2cm$ is injected into the system and the trajectories of the intermediate points are recorded as soon as they enter the trocar port. Figure 6 reports the results of the experiment showing the demonstrations and the distance of the intermediate points from the desired trajectory. The points are kept on the trajectory only where the variability is lower, while noise is not counteracted in the central part of the trajectory, where more variability is allowed.

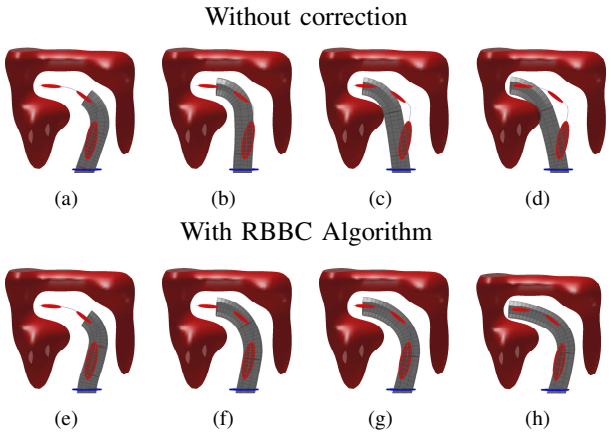

Fig. 4. Comparison between the motion of the robot without correction in the null space (top row) and the same motion when corrections are learnt from demonstrations (bottom row). The figures in the same column correspond to the same time step.

V. DISCUSSION

A critical challenge in the the current approach is to provide multiple demonstrations to obtain the variability information. The experiments of this paper were performed using multiple demonstrations inside a virtual environment, with the focus of the exploitation of the variability information to implement a variability-priority null space algorithm. The proposed algorithm relies on a generic GMM encoding of the task. In future work, more realistic ways of providing this model within a surgical scenario will be explored.

Discussions with surgeons singled out that it is impossible to use pre-operational images, since the geometry of the organs changes during the surgical procedure: organs are actually displaced from the original position to make space for the surgical tools.

The current plan to overcome this difficulty is to build the GMM incrementally in an online manner as the tip of

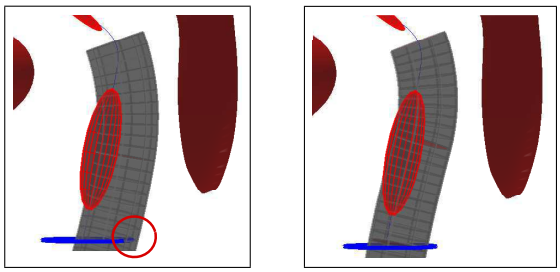

Fig. 5. Demonstration of the usefulness of the proposed minimal intervention approach by comparing the behaviour of two different controllers at the same time step. Left: the robot is fully controlled to have its body tracking the trajectory. Since it is not possible to satisfy all the constraints, a bad compromise is used and the robot moves away from the trocar port tracking error (red circle). Right: the robot is minimally controlled with the RBBC algorithm.

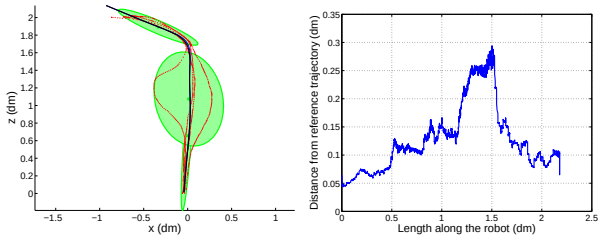

Fig. 6. Disturbance rejection experiment. Left: demonstrations with encoded GMM. Right: distance of the intermediate points of the STIFF-FLOP manipulator from the desired trajectory as a function of the length of the trajectory. The plot represents the average over 10 runs.

the STIFF-FLOP robot enters the body. In this sense, the exploration of all the available space inside the body with the tip could be exploited as it gradually enters into new areas and move the organs aside. The surgeon could even specify high priority viapoints where hard constraints need to be fulfilled, or soft viapoints, where a low covariance should be used. The incrementally built model will be then applied to the body of the robot, as it enters through the trocar port.

Another important issue concerns the importance rating of the constraint during the task. In our experiment, the tip of the robot was considered as freely controllable by the surgeon, and only the null space of the robot was subject to actions. On the other hand, competing constraints could appear during the task and the free motion of the tip could prevent the robot to be kept near the envisaged trajectory where needed.

All the learning was performed at the path level. Since the trajectory is defined once and for all during the task, this is not a big limitation and no generalization to nearby trajectories is needed. More robust solutions based on statistical dynamical systems [11] will be alternatively explored in future work.

The experiment regarding disturbance rejection is only a first prototype of possible uses of the RBBC algorithm for controlling the STIFF-FLOP manipulator. The active rejection used in this paper simply consisted in an additional displacement given to the intermediate points to compensate for the noise. A more accurate study involving the low-level controllers (that are being presently implemented) will be carried out to make the procedure fully operational.

VI. CONCLUSION

This paper shows how Gaussian Mixture Regression can be used to obtain a positional null-space controller for a surgical flexible robot. The resulting variability-priority controller exploits the variability information of the task, demonstrated by an expert user, to learn the desired policy.

In order to comply with the needs of the surgical applications, where the velocity of the task is given by the motion of the human operator, a time-invariant Gaussian mixture regression mechanism, with inputs based on curvilinear distance, was proposed.

The controller represents an implementation of the minimal intervention principle, that corrects the position of the robot only when needed, keeping it within the intrinsic variability of the task. This behaviour proved to be useful when an excessive tight controller would impose a set of commands on the manipulator that its geometry cannot fit.

The controller was succesfully used in simulation to move the body of the robot within the given constraints and to implement a first disturbance rejection scenario for the low level controllers.

REFERENCES

[1] M. Cianchetti, T. Ranzani, G. Gerboni, I. De Falco, C. Laschi, and A. Menciassi, "STIFF-FLOP surgical manipulator: mechanical design and experimental characterization of the single module," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013, pp. 3567–3581.

[2] A. Jiang, G. Xynogalas, P. Dasgupta, K. Althoefer, and T. Nanayakkara, "Design of a variable stiffness flexible manipulator with composite granular jamming and membrane coupling," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 2922–2927.

[3] A. Jiang, A. Ataollahi, K. Althoefer, P. Dasgupta, and T. Nanayakkara, "A variable stiffness joint by granular jamming," in *Proceedings of the ASME 2012 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE 2012*. ASME, 2012.

[4] C. Towell, M. Howard, and S. Vijayakumar, "Learning nullspace policies," in *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, Taiwan, 2010.

[5] A. A. Rajiv Ranganatan and F. A. Mussa-Ivaldi, "Learning to be lazy, exploiting redundancy in a novel task to minimize movement-related effort," *Journal of Neuroscience*, vol. 33, no. 7, pp. 2754–2760, 2013.

[6] Z. Ghahramani and M. I. Jordan, "Supervised learning from incomplete data via an EM approach," in *Advances in Neural Information Processing Systems*, J. D. Cowan, G. Tesauro, and J. Alspector, Eds., vol. 6. Morgan Kaufmann Publishers, Inc., 1994, pp. 120–127.

[7] M. J. Gielniak, C. K. Liu, and A. L. Thomaz, "Task-aware variations in robot motion," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 3921–3927.

[8] S. Calinon, F. D'halluin, E. L. Sauser, D. G. Caldwell, and A. G. Billard, "Learning and reproduction of gestures by imitation: An approach based on hidden Markov model and Gaussian mixture regression," *IEEE Robotics and Automation Magazine*, vol. 17, no. 2, pp. 44–54, June 2010.

[9] S. M. Khansari-Zadeh and A. Billard, "Learning stable non-linear dynamical systems with Gaussian mixture models," *IEEE Trans. on Robotics*, vol. 27, no. 5, pp. 943–957, 2011.

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

[11] S. Calinon, Z. Li, T. Alizadeh, N. G. Tsagarakis, and D. G. Caldwell, "Statistical dynamical systems for skills acquisition in humanoids," in *Proc. IEEE Intl Conf. on Humanoid Robots (Humanoids)*, Osaka, Japan, 2012, pp. 323–329.

[12] S. Calinon, F. Guenter, and A. Billard, "On learning, representing and generalizing a task in a humanoid robot," *IEEE Trans. on Systems, Man and Cybernetics, Part B*, vol. 37, no. 2, pp. 286–298, 2007.

[13] J. Peters and S. Schaal, "Learning to control in operational space," pp. 197–212, 2008. [Online]. Available: http://www-clmc.usc.edu/publications/P/peters-IJRR2008.pdf

[14] E. Todorov and M. I. Jordan, "A minimal intervention principle for coordinated movement," in *Advances in Neural Information Processing Systems (NIPS)*, 2002, pp. 27–34.

[15] M. S. Malekzadeh, D. Bruno, S. Calinon, T. Nanayakkara, and D. G. Caldwell, "Skills transfer across dissimilar robots by learning context-dependent rewards," in *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, Tokyo, Japan, November 2013.

[16] R. J. Webster III and B. A. Jones, "Design and kinematic modeling of constant curvature continuum robots: A review," pp. 1661–1683, 2010.

[17] O. Khatib, "A unified approach for motion and force control of robot manipulators," *IEEE Journal of Robotics and Automation*, vol. 3, no. 1, pp. 43–53, 1987.