Learning autonomous behaviours for the body of a flexible surgical robot*

Danilo Bruno[1] Sylvain Calinon[2,1] Darwin G. Caldwell[1]

Abstract

This paper presents a novel strategy to learn a positional controller for the body of a flexible surgical manipulator used for Minimally Invasive Surgery. The manipulator is developed within the STIFF-FLOP European project and is targeted for a laparoscopic use in remote areas of the abdominal region that are not easily accessible by means of currently available rigid tools. While the surgeon controls the end-effector during the task, the flexible body of the manipulator needs to be displaced to enter inside constrained spaces by efficiently exploiting its flexibility, without touching vital organs and structures. The proposed algorithm exploits the instruments of Machine Learning within the Programming by Demonstrations paradigm to produce a statistical model of the natural movements of the surgeon during the task. The gathered information is then reused to determine a controller in the null space of the robot that does not interfere with the surgeon task and displaces the robot body within the available space in a fully automated manner.

1 Introduction

Minimally invasive surgery is becoming a standard for surgical applications nowadays, since it has the advantage of reducing the post-operatory recovery of patients and is less traumatic for the body. When referred to abdominal applications, they are called laparoscopic surgical procedures and they can be performed either manually or with the aid of consolidated robotic platforms, that increase the precision of the surgeon and reduce errors. Nevertheless, the adoption of laparoscopic procedures can be limited by the current technology, that uses rigid tools to access the remote areas where the surgery is performed. This limitation can sometimes prevent some procedure to be carried out laparoscopically, or make it technically demanding.

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 (Cianchetti et al., 2013; Jiang et al., 2012b,a). The flexibility of the robot allows the surgeon to move within organs to reach remote areas that cannot be accessed easily with rigid tools.

One of the issues that comes with the use of a flexible tool is the control of the body of the manipulator, that does not simply follow the movements of the surgeon, but is dragged along by the end effector following its own dynamics. The STIFF-FLOP robot is therefore actuated to have the possibility of displacing the flexible body to pass around organs and reach for more remote areas. The control of these additional degrees of freedom cannot be performed by the surgeon, who needs to concentrate on controlling the end-effector to perform the procedure, but has to be automated: the resulting controller should be able to drive the body of the robot without interfering with the surgical task and allowing the robot to reach remote areas that the flexibility now allows.

Most algorithms developed for flexible robots rely on external images of the body of the robot to learn the optimal configuration to perform the task (Lyons et al., 2010; Lobaton et al., 2013). Some of them rely on optimization strategies that are aimed at finding the best configuration to fit inside a very constrained environment, such as blood vessels or natural body channels.

As a matter of fact, the novel STIFF-FLOP architecture requires quite different strategies, since the robot has a different structure and a different actuation. From the structural viewpoint, one of the main differences is placed in the possibility of elongating each module of the manipulator, so that similar configurations can be obtained with different lengths of the single modules. Thus, the full configuration of the system needs to take into account the correct elongation of each module, that needs to be kept within the hardware limits.

Moreover, the different geometry of the robot allows the surgeon to perform new kind of surgeries and require a different set of procedures. First of all, no image of the body of the robot is available to the algorithm and the environment in which it operates is supposed to be changing during the procedure (see Section 3). The task requires the robot to move freely inside the environment in certain phases of the surgical operation and in a very constrained environment in other moments, exploiting the

*This work was partially supported by the STIFF-FLOP European project under contract FP7-ICT-287728. The final version of this publication is available via Springer at http://dx.doi.org/10.1007/s10514-016-9544-6.

[1] Department of Advanced Robotics, Istituto Italiano di Tecnologia (IIT), Via Morego 30, 16163 Genova.

[2] Idiap Research Institute, Rue Marconi 19, CH-1920 Martigny, Switzerland.

possible variability to complete the task. Finally, the only information available to the robot comes from the proprioceptive and force sensors that are able to single out the configuration of the robot and the force applied to it.

In this paper we propose to use the programming by demonstrations paradigm to encode the correct behaviour for the body of the manipulator. This choice is driven by the necessity of coding the correct behaviour for the end-effector in a fast and efficient way, that is able to adapt to the different situations without the need of additional coding. The same code that was employed for running the current algorithm is employed to learn autonomous tasks for the end-effector of the robot. Moreover, the same statistical model can be used to extract information to automate subtasks that need to be performed without additional input from the user.

By using the above sensory input and the movements produced by the surgeon during the task, the proposed algorithm is able to control the body of the flexible surgical robot STIFF-FLOP, by exploiting the redundant degrees of freedom and the history of movements of the end-effector controlled by the surgeon during the operation.

In order to do this, we build a statistical model of the movements of the tip; the information stored in the model can then be reused to calculate what are the best displacement for the body of the robot to keep it within a safe area characterized by the movements of the end-effector previously achieved by the surgeon.

The advantages of the method rely on the fact that the surgeon does not have to worry about the behaviour of the robot, that will naturally follow the tip without interfering with the teleoperation by the surgeon, who always has the full control of the relevant degrees of freedom. The final aim is to provide technological help to bring the surgical tool mounted on the tip in the remote area where it is needed, without adding any extra cognitive load to the human operator.

2 Related work

The current paper proposes a novel approach to learning from demonstration in the context of a surgical robot. The main innovations lie in the use of an online learning algorithm to control the redundancy of the robot within a surgical environment, a biologically inspired optimization strategy to evaluate the null space commands to send to the robot and a novel application of Dirichlet Processes for online clustering.

The use of learning techniques in the null space is usually limited to learn specific policies from demonstrations (Towell et al., 2010; Nordmann et al., 2012) or in order to optimize some reward function to fulfill secondary tasks (Rajiv Ranganatan and Mussa-Ivaldi, 2013; Gielniak et al., 2011). The novelty of the current approach consists of learning the null space policy in an online manner from demonstrations, while the surgeon completes the procedure, and is simultaneously applied to drive the robot and allow the user to complete the task.

The application of learning techniques for surgical robots is quite novel. On one side, the difficulties come with the fact that the surgical context puts obvious limitations to the range of feasible automated tasks. The safety requirements demand that the automation of the procedures meets very rigid standards. Moreover, the robot is primarily meant to be an instrument to be employed by a surgeon, who has a great expertize in practicing very complicated operations that can hardly be replaced by a well trained or programmed robot. For this reason, machine learning is usually applied to some repeated and standard parts of the surgical procedure such as suturing or retracting organs (Van Den Berg et al., 2010) or to surgical image recognition and configuration identification (Reiter et al., 2011). The current paper, instead, uses learning as a tool to make the robot capable of better assisting the surgeon during the operation and allowing him/her to perform operations that would be very difficult by using other tools.

The online learning feature is also the main difference from our previous publication on the topic (Bruno et al., 2014), where several demonstrations of the task to perform were given in advance and used to learn the statistical model whence the null space policy could be obtained. This way of proceeding, even if formally correct, was not easily applicable to a real surgical procedure, because the task is not repeatable and the null space policy was actually needed to perform the demonstrations. The present formulation is instead fully operational within a single surgical operation.

An online clustering algorithm is used to convert the movements of the end-effector that is controlled by the surgeon into a statistical model. There are many algorithms performing online clustering (see, for example (Zhang et al., 2003; Song and Wang, 2005)). Most of these algorithms base the arrangement of the different clusters on a fixed threshold over the likelihood that a new point belongs to the current distribution. Then, a split and merge technique is employed to add or remove clusters, followed by a reorganization with an EM strategy.

Regarding the current work, as we shall see in section 5, the geometry of the robot requires a very precise control over the distance between the different points of the manipulator. Moreover, the flexible nature of the robot allows it to elongate and shrink between precise hardware limits. For those reasons, a threshold based on the distance of the samples instead of the probability is more directly controllable within the current setting.

For this reason, we chose to use a generalization of the DP-Means algorithm (Kulis and Jordan, 2012), that uses the geometric distance as a threshold to evaluate the be-

longing of each sample to the current distribution. In order to adapt it to the current setting, the algorithm was modified to allow it to work incrementally as new points are introduced. Moreover, the incremental evaluation of the parameters of each component is made more precise with respect to the original algorithm by exploiting a MAP estimate at each time step (Gauvain and Lee, 1994). The mix of these two techniques gives rise to an incremental GMM algorithm that results to be fast and efficient for the online use that we need.

Finally, a novel mechanism to encode the motion of the flexible robot is presented, inspired by biological studies on how the octopus moves its arm to perform natural tasks such as reaching and grasping (Flash and Hogan, 1985; Zelman et al., 2013). The approach builds upon the dynamical system formulation of whole body movements presented in (Malekzadeh et al., 2014), that is based on the description of each static pose of the continuum arm as the solution of a second order spring-damper system in the curvilinear distance s.

The same approach is exploited in this paper to describe at each time step the static pose that the robot should have in order to best exploit the variability of the demonstrations provided by the surgeon. The degree of optimality is evaluated by using a Linear Quadratic Regulator (LQR) that is finding a compromise between the curvature attained by the robot and the available space.

3 Surgical flexible robotics

The present section describes in more details the surgical problem under study, together with the challenges that need to be faced. Then a description of the kinematics of the current prototype of STIFF-FLOP manipulator is provided.

3.1 Considered surgical problem

When performing Minimally Invasive Surgical (MIS) procedures, the surgeon inserts the tools inside small openings in the abdomen, called trocar ports, and moves inside the abdominal area to access the surgical site. The abdomen of the patient is inflated to provide more space and some of the organs are eventually displaced from their original position (retracted).

Nevertheless, some surgical procedures require the robot to access more remote areas, that are enclosed between bones or other organs that cannot be easily moved. In this case, the procedure can prove to be impossible to be performed laparoscopically.

One example of these procedures is Total Mesorectal Excission (TME). This procedure aims at removing the final part of the colon of cancer ill patients (Yang et al., 2012). The procedure can be currently done laparoscopically, but the surgeon needs to face several challenges, since he/she has to enter with the tools inside the pelvis area to cut and remove the final tract of the colon. This consists of a fairly cylindrical area, constrained by the pubic bones on each side. The access and the cutting inside the pubic area is technically difficult, since the surgeon is often forced to move aside the tissue he/she needs to cut and even cross the instruments to obtain the necessary tension.

During the procedure, the surgeon controls the end-effector of the robot and performs the medical part of the task, leaving the motion of the whole arm to the control modules. Such control algorithms 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 (Towell et al., 2010; Rajiv Ranganatan and Mussa-Ivaldi, 2013).

In this paper we propose a controller that is able to steer and displace the flexible body, by learning the correct behaviour from the movements that the surgeon performs with the tip during the task. The algorithm treats these movements as demonstrations, extracts the variability information about the task online and exploits this to efficiently move the body of the robot.

With this approach, the surgeon does not need to manually set constraints for the body of the robot while performing the task (for instance in the form of viapoints), where critical situations might occur. In fact, the robot will be forced to pass with the whole body through those points where the surgeon moved the end-effector using a low variability.

3.2 The STIFF-FLOP flexible robot

The European project STIFF-FLOP is aimed at building a surgical flexible robot capable of performing demanding surgical applications such as TME. The robot is presently under development.

The current prototype of the robot is composed of 2 cylindrical sections (links). Each link 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.

Each module can be modeled as a constant curvature section of a circle, see Fig. 1. In its local frame, the rest position (no chamber is inflated) corresponds to the module aligned along the vertical axis $\boldsymbol{e}_3$, with a rest length L_0. The current prototype of the single module is 50 mm long in the rest position and has a diameter of 40 mm. When totally inflated, it can elongate by 80%. Moreover, each link can bend at approximatively 180°. The current prototype does not have surgical dimensions yet and can be used inside phantom environments for testing. The final

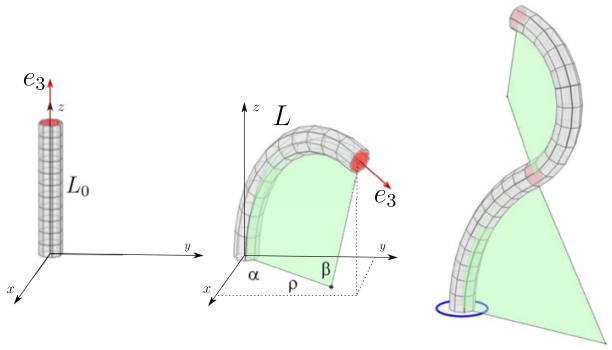

Figure 1: The Inverse Kinematics simulator of the STIFF-FLOP robot. Left: a single module in the rest configuration. Center: a single module modelled as a constant curvature module. Right: the full robot, obtained putting 2 sections together.

version of the robot will then be scaled down to surgical dimensions.

The robot is moved through a positional controller, implemented by using an inverse kinematics model exploiting the constant curvature model of each link, which is detailed in (Calinon et al., 2014b). The employed strategy has the following advantages:

1. it is modular, so that an arbitrary number of constant curvature modules can be added;
2. it is based on Jacobians, so that all the standard tools employed in more conventional robots can be used;
3. it allows us to evaluate the position and orientation of any point along the kinematic chain, so that any point can be displaced if needed.

4 Statistical learning and control approach

In this section, we present the mathematical tools that will be used to develop the algorithm. The content of this section can be summarized as follows:

1. a brief survey of the Learning from Demonstration paradigm, with some highlights about the way it is employed to learn the correct behaviour for the body of the robot inside a surgical scenario;
2. an introduction to a novel use of Dirichlet Process for online clustering;
3. a short review of optimal control in linear quadratic system.

4.1 Learning from demonstrations

We will use the Learning from Demonstrations (LfD) paradigm to extract constraints for the motion of the body of the flexible STIFF-FLOP robot.

Our aim is to use demonstration of the task provided by an expert user to extract a statistical model of the task, which can then be used for reproduction under different situations or with different architectures.

In this paper this is provided by encoding the demonstrations into a Gaussian Mixture Model (GMM), which represents the probability that a given datapoint belongs to the distribution of demonstrations. This statistical model can be used together with a regression technique that estimates a set of output variables in term of another set of input variables by conditional probability evaluation. The details are provided in Appendix A.

The use of LfD to learn the behaviour of the body of the flexible robot during a surgical procedure offers a unique challenge, due to the fact that the task cannot be demonstrated in advance without the envisaged body controller, since the body of the robot needs to be displaced while the task is performed. It is also impossible to provide several demonstrations in a batch manner, since the surgical procedure is not repeatable. Finally, the pre-operatory images cannot be used to learn the task offline before the actual procedure, since the geometry of the abdomen changes significantly, due to the inflation and the retraction of organs.

In this paper we solve this challenge by learning the statistical model incrementally as the tip is moving inside the body, by considering its movements as the expert demonstrations, and applying the resulting learnt controller to the robot body, as it enters inside the patient body.

Learning occurs in the 3-dimensional Cartesian space where the samples of the end-effector positions are collected, augmented with the curvilinear distance from the trocar port measured along the manipulator as $\boldsymbol{\xi}_n = [\boldsymbol{\xi}_n^I, \boldsymbol{\xi}_n^O]^\top$. The indices $_I$ and $_O$ label the input and output part of the vector corresponding to the distance from trocar variable and the Cartesian position respectively.

The model of the motion of the end-effector is represented by a GMM whose components have mean and a covariance matrix of the form:

$$\boldsymbol{\mu}_i = \begin{bmatrix} \boldsymbol{\mu}_i^I \\ \boldsymbol{\mu}_i^O \end{bmatrix}, \quad \boldsymbol{\Sigma}_i = \begin{bmatrix} \boldsymbol{\Sigma}_i^I & \boldsymbol{\Sigma}_i^{IO} \\ \boldsymbol{\Sigma}_i^{OI} & \boldsymbol{\Sigma}_i^O \end{bmatrix}. \tag{1}$$

As we shall see in Section 5, this choice allows us to control the length of each module effectively, by exploiting the length information contained in each datapoint as input for the regression process during reproduction (see Appendix A).

4.2 Online Dirichlet Process Clustering

The considered application requires points to be added to the statistical model online. For this reason, we chose to exploit a variant of Dirichlet Process clustering, called

DP-Means, which can be also used for online applications (Kulis and Jordan, 2012).

Dirichlet process is a common Bayesian non-parametric approach to cluster data by automatically estimating the number of components required for the process. Since it needs Gibbs sampling to produce the clustering, it cannot be generally used to encode a GMM online. The DP-Means variant of Dirichlet process that we will use was initially introduced to simplify the learning process by making a parallel with the K-Means algorithm (MacQueen, 1967) by small variance asymptotics analysis.

The idea is to substitute the evaluation of the posterior probabilities of each component with the distance of datapoints from the center of each component, thus avoiding the use of Gibbs sampling techniques. This can be done by specifying a maximum distance between two components: when this is reached, a new component is automatically created.

The regression from Gaussian Mixture Models extracts a piecewise linear trajectory from the model, requiring GMM with a sufficient granularity (represented by the number of components) to encode the correct behaviour for the different modules. For this reason, we set the splitting distance to the rest length of the single module, to be sure that each module is covered by more than 1 component when elongated.

Starting from an initial configuration of a GMM with K components, each point $\boldsymbol{P}$ is assigned to the component whose center is at a lower distance. In any case, if the distance from any component is higher than a given threshold λ, the point is assigned to a new component, having $\boldsymbol{P}$ as center and a preassigned fixed covariance. After all the assignments have been performed, the eventual empty clusters are discarded and new values for centers and covariance of the remaining clusters are reevaluated by using a MAP estimate taking the previous values of center and covariance as prior information (Gauvain and Lee, 1994). The whole procedure is repeated a sufficient number of times, until a stable configuration is reached.

In our case, the algorithm has to be adapted to the fact that a new point is added at each time step. This means that no iteration on the points is needed, since only the last one needs to be assigned each time. Moreover, no iteration of the whole algorithm is needed, since the stability is reached at the first iteration. This produces a very fast clustering algorithm, that only needs a splitting distance threshold as input.

The full algorithm is summarized in Algorithm 1.

4.3 Optimal control with Linear Quadratic Regulator

The extraction of the data from the model is performed by Gaussian Mixture Regression (GMR). This is performed

Algorithm 1 Online DP-Means algorithm. The superscripts (n) and (o) stand for new and old respectively.

Require: Streaming flow of points $\boldsymbol{x}_t$ at each time step
Require: Splitting length threshold λ
Require: Minimum covariance $\hat{\boldsymbol{\Sigma}}$
Ensure: GMM encoding the points
  Initialize GMM $K = 1, \boldsymbol{\mu}_1 = \boldsymbol{x}_0\ ,\ \boldsymbol{\Sigma}_1 = \hat{\boldsymbol{\Sigma}}\ ,\ \pi_1 = 1$
  repeat
    $N \leftarrow$ total number of points until current time t
    for $k = 1 \rightarrow K$ **do**

$$\begin{aligned} d_k &= dist(\boldsymbol{x}_t, \boldsymbol{\mu}_k) \\ d_{K+1} &= \lambda \end{aligned} \tag{2}$$

    end for
    assign $\boldsymbol{x}_t$ to component $k^* = \arg\min_k(d_k)$
    if $k^* = K + 1$ **then**
      $K = K + 1, \boldsymbol{\mu}_{k^*} = \boldsymbol{x}_t\ ,\ \boldsymbol{\Sigma}_{k^*} = \hat{\boldsymbol{\Sigma}}\ ,\ \pi_{k^*} = \frac{1}{k^*}$
    else

$$\begin{aligned} \pi^* =& \frac{1}{N} + \pi_{k^*} \quad , \quad \pi_{k^*} = \frac{1}{K} \\ \boldsymbol{\mu}_{k^*}^{(n)} =& \frac{1}{\pi^*}\left(\pi_{k^*}\boldsymbol{\mu}_{k^*}^{(o)} + \frac{\boldsymbol{x}_t}{N}\right) \\ \boldsymbol{\Sigma}_{k^*}^{(n)} =& \frac{\pi_{k^*}}{\pi^*}\left(\boldsymbol{\Sigma}_{k^*}^{(o)} + (\boldsymbol{\mu}_{k^*}^{(o)} - \boldsymbol{\mu}_{k^*}^{(n)})(\boldsymbol{\mu}_{k^*}^{(o)} - \boldsymbol{\mu}_{k^*}^{(n)})^\top\right) + \\ & \frac{1}{N\pi^*}\left(\hat{\boldsymbol{\Sigma}} + (\boldsymbol{x}_t - \boldsymbol{\mu}_{k^*}^{(n)})(\boldsymbol{x}_t - \boldsymbol{\mu}_{k^*}^{(n)})^\top\right) \end{aligned}$$

    end if
    renormalize $\boldsymbol{\pi}$
  until end of points

by splitting the variables into two sets (input and output) and evaluating the conditional probability of the output variables over the input ones. The conditional mean and covariance of the resulting distribution represent a weighted least square estimate of the output together with the error over the estimate.

In the present case, we are using the distance s from the trocar port as input variable for the model described in eq.(1), thus getting the most probable Cartesian position associated to the given distance as output. For each value of s we therefore obtain a trajectory $\hat{\boldsymbol{\mu}}_s^O$ and the error on the trajectory $\hat{\boldsymbol{\Sigma}}_s^O$, as explained in Appendix A.

The reproduction step of GMR can be improved by coupling a simple second order linear dynamical system to the reproduction step. This improves the smoothness of the reproduced trajectory and allows some additional optimality criterion to be fulfilled (Calinon et al., 2014a).

In particular, since we are not working with straight links, a precise control of some points along the manipulator does not prevent each section from deviating from the desired trajectory because of a too high curvature (see Section 6). For this reason, an optimal control strategy will be used, to obtain a compromise between optimal tracking and curvature minimization, by minimizing the cost function

$$c = \frac{1}{2}\sum_{s=0}^{K}\Big(\boldsymbol{u}_s^\top \boldsymbol{R}\boldsymbol{u}_s + (\boldsymbol{x}_s - \boldsymbol{x}_s^*)^\top \boldsymbol{Q}_s(\boldsymbol{x}_s - \boldsymbol{x}_s^*)\Big), \qquad (3)$$

for a linear dynamical system of the form

$$\begin{pmatrix}\dot{\boldsymbol{x}}\\ \ddot{\boldsymbol{x}}\end{pmatrix} = \underbrace{\begin{pmatrix}\boldsymbol{0} & \boldsymbol{I}\\ \boldsymbol{0} & \boldsymbol{0}\end{pmatrix}}_{\boldsymbol{A}}\begin{pmatrix}\boldsymbol{x}\\ \dot{\boldsymbol{x}}\end{pmatrix} + \underbrace{\begin{pmatrix}\boldsymbol{0}\\ \boldsymbol{I}\end{pmatrix}}_{\boldsymbol{B}}\boldsymbol{u}, \qquad (4)$$

where the dot denotes the derivative with respect to the arc length s.

As explained in Section 2, this approach is driven by a biological inspiration and reveals to be useful in the robotics context, since the curvature of the robot is directly proportional to the second derivative of the displacement with respect to the arc length.

The value of $\boldsymbol{R}$ selects the trade off between tracking precision of the final trajectory and curvature minimization. The choice of $\boldsymbol{R}$ is performed by the user and depends on the system in use. For the robot considered, it has been chosen as a constant diagonal matrix $0.1\boldsymbol{I}$ and does not need to be adapted to the different situations.

In order to learn the controller from the demonstrations, by taking their variability into account, the cost function is set by extracting the desired shape $\boldsymbol{x}_s^*$ and the position error $\boldsymbol{Q}_s$ from demonstrations by using GMR (eq.(16)), i.e.

$$\boldsymbol{x}_s^* = \hat{\boldsymbol{\mu}}_s^O \quad , \quad \boldsymbol{Q}_s = (\hat{\boldsymbol{\Sigma}}_s^O)^{-1}. \qquad (5)$$

Under the hypothesis that the control command $\boldsymbol{u}$ is proportional to the state variable $\boldsymbol{x}$ (linear feedback), a solution to the above problem is determined by a variant of the Linear Quadratic Regulator (LQR) controller in the form

$$\boldsymbol{u}_s = -\boldsymbol{R}^{-1}\boldsymbol{B}^\top \boldsymbol{S}_s\begin{pmatrix}\boldsymbol{x}_s - \boldsymbol{x}_s^*\\ \dot{\boldsymbol{x}}_{\boldsymbol{s}}\end{pmatrix} + \boldsymbol{R}^{-1}\boldsymbol{B}^\top \boldsymbol{d}_s. \qquad (6)$$

The feedback matrix $\boldsymbol{S}_s$ and the feedforward vector $\boldsymbol{d}_s$ are the solution of the following differential equations

$$\dot{\boldsymbol{S}}_s + \boldsymbol{A}^\top \boldsymbol{S}_s + \boldsymbol{Q} + \boldsymbol{S}_s\boldsymbol{A} - \boldsymbol{S}_s\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^\top \boldsymbol{S}_s = 0. \qquad (7)$$

$$\dot{\boldsymbol{d}}_s = -\boldsymbol{A}^\top \boldsymbol{d}_s + \boldsymbol{S}_s\boldsymbol{A}\boldsymbol{x}_s^* + \boldsymbol{S}_s\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}\boldsymbol{d}_s - \boldsymbol{S}_s\dot{\boldsymbol{x}}_s^*. \qquad (8)$$

Notice that eq.(7) is identical to the classical Riccati equation solving the LQR problem with fixed target; the difference is placed in the addition of eq.(8), representing a feedforward term $\boldsymbol{d}_s$.

The optimal trajectory can be obtained by integrating eqs.(7) and (8) starting from a given initial or final condition, depending on the problem under study. Usually, for slowly varying dynamics, the problem is converted into an infinite horizon scenario, where eq.(8) is dropped and eq.(7) is considered constant at each time step. The problem is then reduced to determining at each time step the solution of the algebraic Riccati equation

$$\boldsymbol{A}^\top \boldsymbol{S}_s + \boldsymbol{Q}_s + \boldsymbol{S}_s\boldsymbol{A} - \boldsymbol{S}_s\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^\top \boldsymbol{S}_s = 0. \qquad (9)$$

Once $\boldsymbol{S}_s$ has been determined, the trajectory can be evaluated by integrating the dynamical system (4). We will denote by $LQR(\boldsymbol{x}_s^*, \boldsymbol{\Sigma}_s)$ the points of the integrated trajectory at each time step.

5 Problem setting

The learning problem we want to tackle is that of learning the optimal nullspace displacement that is needed to move the body of the flexible STIFF-FLOP robot while the surgeon is performing the procedure. Those motions should be safe and move the robot only inside the areas that the end-effector of the robot already explored, meaning that they are accessible to the whole manipulator.

In order to avoid unnecessary movements, the body will not be forced to follow the same trajectory of the end-effector, but will be guided to move within the variability boundaries extracted from the demonstrations provided by the teleoperation of the end-effector.

The main difference in the architecture of the STIFF-FLOP robot is placed in the possibility of changing the length of the single modules in a given elongation range. While inserting the robot inside the trocar port and while

manoeuvring it, the elongation should be controlled so that the body of the robot and its base follow the end effector smoothly. The risk is that the evolution of the end effector forces some of the modules to elongate more than others and reach the hardware limits. The solution provided by weighting the Inverse Kinematics is not necessarily good, because of the non linearities of the system that make the correct weighting strictly dependent on the configuration.

For this reason, we need an algorithm that is able to set the correct length of each section of the robot at every time step. In this case, the learning approach that is employed in this paper allows us to input a desired length, that can also vary along the task, and obtain a desired pose for the robot as a result of a regression procedure, thus providing a very effective control. In this paper we choose to set the length of each section of the manipulator in the middle of the range of possible elongation, so as to keep far from the hardware limitations (40% of elongation).

The proposed learning algorithm needs to perform three consecutive actions at each time step:

1. encode the motion of the tip into a GMM by using the DP-Means algorithm at each time step;
2. extract the mean trajectory of the end-effector together with the variability information;
3. use the information to calculate a displacement for selected body points along the manipulator and convert them into a single null space command.

The choice of using a GMM to encode the free space comes from the observation that a statistical model of the points reached by the surgeon with the end–effector can be used to represent the probability of being placed in an allowed position, defined by the absence of organs or obstacles.

The key idea of the algorithm is to encode at each time step the position of the tip together with its distance from the trocar port measured along the manipulator. The statistical model constructed on this augmented set allows us to extract the correlation between these variables along the task.

As a consequence, the distance variable can be exploited as a regression input for the algorithm, to obtain at each time step the desired position and variability of all the points along the manipulator, since we can estimate their desired distance from the trocar port, as we shall see in the following section.

Another important point is concerns the variability information within the current context. In fact, it is not meant as being a measure of the errors made by the expert surgeon (all demonstrations are expected to be correct executions of the task). In contrast, variability is exploited to encode the free space within the surgical field and the surgeon should be aware of this feature of the tool while operating it. For this reason the surgeon is required to explore the free space to show which areas can be exploited to perform the task. Since this exploration phase usually takes place naturally in the surgical procedure, it can be incorporated into the steps of a protocol to follow when a STIFF-FLOP robot is employed. In this respect, an expert surgeon will be the one who is able to make a more efficient use of covariance (Sternad et al. (2011)) to provide the most useful information to the learning algorithm.

5.1 Null space controller

After inserting the robot inside the trocar port, the surgeon starts controlling the end-effector, moving it inside the abdomen to perform the surgical procedure. At each time step, the current position $\boldsymbol{x}$ of the end-effector is recorded, together with the distance l of the end-effector from the trocar port measured along the manipulator. The geometry of the robot makes this measurement quite straightforward, since it is possible to obtain the actual length of each module from the joint coordinates, by using the constant curvature hypothesis as in (Calinon et al., 2014b). Each datapoint $[L, \boldsymbol{P}^\top]^\top$ is incrementally encoded into a GMM by using the online DP-Means algorithm. The splitting length threshold λ (see eq. (2)) is set as the rest length L_0 of each module of the STIFF-FLOP robot. This allows us to have at least one component of the GMM taking care of the points of a module, thus allowing a sufficient granularity for the precision that we want to achieve.

The GMM model is initialized at the starting position of the robot, that is also used to identify the position of the trocar port. This is performed by putting the end-effector of the robot at the entry point and defining its current frame as the trocar reference frame. Then, a discretization of the backbone of the robot is performed and the points used to encode an initial GMM with K components, where K is the same as the number of modules of the robot. This GMM is regularized by adding a constant covariance term $\Sigma_0 = r_0 \boldsymbol{I}_4$, where r_0 is the radius of the cylindrical section of the manipulator.

This choice allows us to impose the trocar position and to enforce the fact that the space is still unexplored. During the initial time steps, the body of the robot will stay confined inside the region represented by the initial GMM.

At each time step, the tip positions are then encoded into the GMM

$$\boldsymbol{y} = [L, \boldsymbol{P}^\top]^\top \sim \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{y}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{10}$$

where K is the variable number of components, that incrementally varies as new points are added. As we shall see, the length information can be used to learn a desired elongation for each module.

Algorithm 2 Initialization algorithm for exploration GMM

Require: Initial configuration $\boldsymbol{q}$ of the robot
Require: Radius of the manipulator r_0
Ensure: $\pi_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i$ GMM representing the initial explored area
  $K \leftarrow$ number of modules
  $s \leftarrow [0 \dots N]$ vector of N arm indexes
  $\boldsymbol{\Sigma}_0 \leftarrow r_0 \boldsymbol{I}_4$
  for $n = 1 \rightarrow N$ **do**
    $\boldsymbol{P}_n \leftarrow$ position of point at arm index $s(n)$
    $L_n \leftarrow$ distance from tip along manipulator of point at arm index $s(n)$
    $\boldsymbol{\xi}_n \leftarrow [L_n, \boldsymbol{P}_n^\top]^\top$
  end for
  $\pi_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i \leftarrow \text{GMM}(\boldsymbol{\xi}_1 \dots \boldsymbol{\xi}_N)$ with regularization $\boldsymbol{\Sigma}_0$

For each of the selected points $\boldsymbol{P}^*$ that we want to control along the manipulator, the desired distance from the trocar port L^* is given as an input to the algorithm. In this paper, this number is chosen to keep the length of each module in the middle of the possible range, in order to improve manipulability. The details of the calculation are reported in Alg. 3.

A desired position $\boldsymbol{\mu}^*$ is evaluated for the point $\boldsymbol{P}^*$ by using Gaussian Mixture Regression (GMR), having L^* as input, together with variability information in the form of a covariance matrix $\boldsymbol{\Sigma}^*$. This information is used to evaluate a displacement command $\boldsymbol{V}(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*)$.

In the present paper, the desired position of $\boldsymbol{P}^*$ is evaluated as a compromise between tracking precision (within the extracted covariance $\boldsymbol{\Sigma}^*$) and curvature minimization, by using the LQR approach with infinite horizon, as described in Section 4.3. The displacement vector for $\boldsymbol{P}^*$ is evaluated as

$$\boldsymbol{V}(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*) = \boldsymbol{P}^* - LQR(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*). \tag{11}$$

The graphical version is represented in Fig. 2.

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. Each displacement is then projected in the null space of the inverse kinematics.

Finally the joint displacements are merged into a unique command to the robot, by making the average of the single joint movements for the single points. This average procedure is used to implement a minimal intervention principle (Todorov and Jordan (2002)): the more relevant constraints are weighted as having a higher importance and correspond to a bigger displacement.

The full controller Robot Body Behaviour Control (RBBC) is summarized in Algorithm 3.

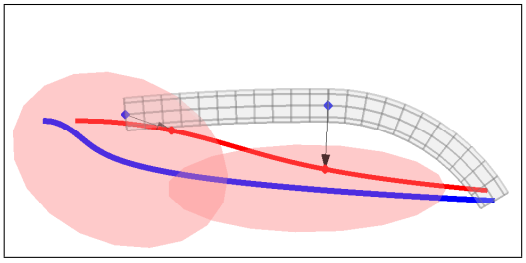

Figure 2: The calculation of displacement. A minimal curvature trajectory (in red) is calculated by using the LQR approach and the new attractor positions are evaluated on this trajectory. The distance between the attractor points (in red) are specified to lie in the middle of the elongation range of the robot. The blue line represents the trajectory evaluated by GMR.

The covariance information is taken into account by appropriately weighting the displacement vector moving the point $\boldsymbol{P}^*$ towards the desired position. As a result, if the point is within the extracted covariance, a very small displacement is performed, while the full displacement is used as the point goes farther from the desired error.

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 in each point. If the variability is high, the robot is allowed to move freely away from the trajectory (if this is needed to perform the task), while the robot is attracted towards the trajectory where the variability is lower.

A second important aspect regards disturbance rejection. During the task, the presence of noise can move the manipulator away from the desired trajectory. If the variability is high, the robot does not need to compensate for it, since it does not affect the task. On the other hand, the robot needs to be attracted towards the assigned trajectory whenever the noise pushes it away from the safe area defined by the initial demonstrations. The proposed approach implements a Minimal Intervention Principle over the whole controller, since the joint commands that are moving the body the most will have a higher importance, while the other points are left almost undisturbed.

6 Experiments

In the following section we present experiments performed with a simulator of the STIFF-FLOP robot in Matlab and with the physics simulator of the robot developed within the project. Both simulators use the Inverse Kinematics discussed in Section 3.2, that is the same implemented in the prototype of the robot. The full algorithm is developed

in the 3D environment. The value of R is experimentally chosen and fixed to $R = 0.1$.

The use of the Matlab simulator allows to best evaluate the performance of the robot and extract graphs and numerical quantities.

6.1 Motion in a constrained environment

Within the Matlab simulated environment, the organs are represented by ellipsoids placed in front of the trocar port. The ellipsoids are placed in order to simulate a constrained environment where the robot should be inserted to perform a surgical procedure. The use of geometrical shapes allows to us to evaluate the performance of the system by computing the distance of the robot from obstacles and the eventual collision.

After entering the body through the trocar port, the STIFF-FLOP robot is moved around in front of the trocar port, to show the free space that can be exploited in the movement (Fig. 3b-c). Then, the robot is moved inside a constrained environment, simulating a possible surgical scenario.

In order to be able to reach the remote areas inside the pelvis, the STIFF-FLOP robot needs to be mounted on the tip of a standard position-controlled manipulator. This allows a precise control of the position of the base of the STIFF-FLOP robot inside the body.

In order to take this structure into account, an additional constraint is added, by rotating the base frame towards the trocar port when the robot is inside the abdominal region. This constraint makes the current experiment similar to the situation that the robot will be facing in a surgical application.

As we can see in Fig.3, the robot can be moved by controlling the tip, while passing between organs inside a very constrained area, without pressing on nearby organs. During the task, the body of the robot is bent along a trajectory that is less curved than the average trajectory (in red) and passes on the border of the available free space.

In order to evaluate the method quantitatively we measured how the algorithm is able to keep the body of the robot inside the demonstrated space, avoiding collisions with the obstacles. Moreover, we evaluated how the method is effective in keeping the length close to the required values.

In order to provide a baseline for comparison, we implemented a simple algorithm which will be referred to as closest point algorithm. Instead of building a statistical model of the demonstrated points, we directly take the Cartesian positions of the points reached by the surgeon and augment them with the corresponding distance from the trocar port. At each time step, we attract each chosen control point to the closest demonstrated point. The distance is evaluated in the augmented $4-$dimensional space;

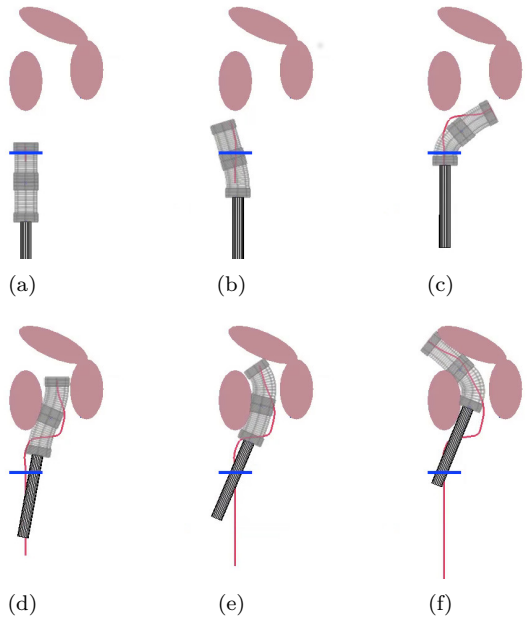

Figure 3: The robot is moved inside the operating environment, passing near organs to reach remote area and perform the surgical task. The body of the robot is displaced to stay confined within the demonstrated variability around the average trajectory, represented in red. The position of the virtual attractors is automatically evaluated as a compromise between tracking precision and minimization of curvature. The blue line represents the position of the trocar port.

this allows us to attract the robot towards the set of points corresponding to the desired distance from the trocar port, in order to take the desired length of each module into account. In this example, no LQR controller was employed, since no information on variability is available (no statistical model is constructed).

Finally, in order to tone down, in the analysis, the potential bias introduced by the user demonstrating the task, we performed a further experiment after the insertion phase is complete. After the robot reaches the target destination inside the constrained environment, a series of random non-colliding tip motions are performed and the response of the robot is evaluated.

The results of this comparison are the following:

1. the RBBC algorithms effectively keeps the robot at the desired distance from the obstacles with a similar behaviour in the approach and the random motion phase (see Fig.5(Left) for the random phase);
2. the closest point algorithm has a similar average behaviour as the RBBC algorithm, both in the insertion and random phase, as shown in 5(Right) for the random phase;
3. the biggest difference is placed in handling the length of the modules, as shown in Fig.6; the RBBC algorithm is able to keep the ratio between the two segments of the robot close to 1, and the length inside the hardware limits; on the other hand, using the closest point algorithm the control over the length is not precise, resulting into ranges that reach the hardware limitations of the robot;
4. the collision probability for the conducted experiments is zero for the RBBC algorithm, both in the insertion task and random motion phase; it raises to 5% for the closest point algorithm during the approach phase and 14% for the closest point in the random phase.

The better behaviour of the RBBC algorithm is mostly placed in the minimization of the curvature of the manipulator by using the LQR approach. In fact, the collisions with the closest point algorithm happen in all cases with the uncontrolled points placed at $s = 0.5$ and $s = 1.5$, because of a too high curvature. This fact also pushes the robot to elongate too much to keep the correct distance between the control points, thus reaching the hardware limits. A hand-tuned LQR approach could eventually improve the closest point algorithm, providing an ad hoc solution for each different task. The proposed algorithm has the advantage of providing an automated tuning of the LQR parameters varying along the task, learnt online from the surgeon demonstrations.

We finally estimated the effect of the $\boldsymbol{R}$ factor on the performance of the algorithm. This allows us to show that the good perfomance is not due to the fine tuning of the parameter, but to the effect of the minimal intervention

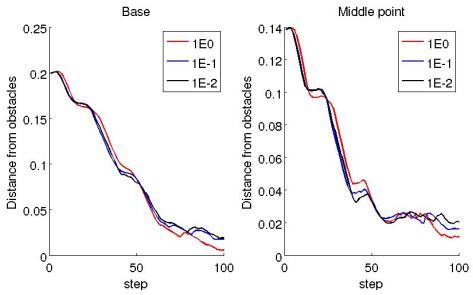

Figure 4: The distance of the obstacle of the base (Left) and middle point (Right) of the STIFF-FLOP robot during the insertion task with different values of R. As we can see, there is a slight difference in the achieved distance, which is higher with lower values of R, since the tajectory is better tracked. But the effect of R is very small and can be used as a confidence parameter to fine tune the distance from the obstacles.

principle driving the controller. For this reason, we repeated the insertion experiments with different values of $\boldsymbol{R}$, ranging from $\boldsymbol{R} = \boldsymbol{I}$ to $\boldsymbol{R} = \boldsymbol{I}0.01$ and evaluated the achieved distance of $\boldsymbol{R}$ from the obstacles (see Fig.4). The results show that the effect of $\boldsymbol{R}$ is small within a very large interval (2 orders of magnitude).

6.2 Retraction of the STIFF-FLOP robot

After the surgical operation is over, the robot needs to be retracted back from the surgical field and extracted from the body. This is a quite easy operation for a rigid tool, but it can become difficult with a flexible surgical tool that is working between soft organs.

The current algorithm allows us to automatically extract the robot back in a safe way by moving the tip along the average estimated path that was previously used as a target trajectory, by taking into account variability information. First the tip of the robot is aligned with the vector tangent to the trajectory. Then, the trajectory is followed in the opposite direction keeping the orientation aligned along the tangent vector at each time step. The result is shown in Fig. 7.

In this experiment, LQR is similarly used to calculate the retraction trajectory, represented in blue in Fig.(7). This choice results in a smoother rejection that avoids oscillations of the tip.

6.3 Test on the simulator of the real platform

This experiment is aimed at showing that the proposed algorithm is able to perform well in a 3D environment.

The simulator is constructed for the novel 3 modules prototype and exploits the real physical model of the robot

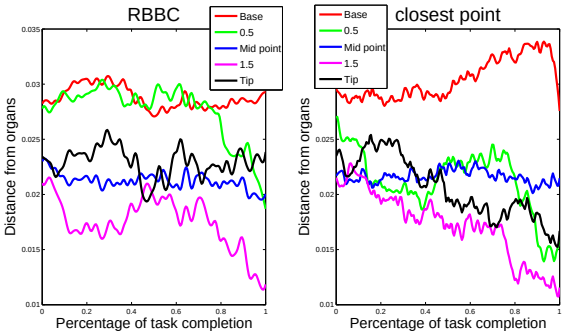

Figure 5: Distance from the obstacles for non colliding random tip motions. *Left :* the RBBC algorithm is used. *Right :* on the right a simpler closest point approach is employed. The x-axis represents the percentage of motion completion and the motion is averaged over 5 different trials.

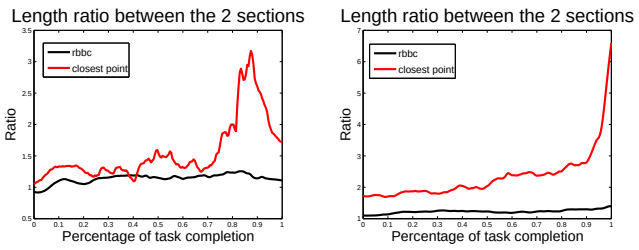

Figure 6: Ratio between the length of the two modules during the tasks. *Left :* the insertion task is shown. *Right :* a non colliding random tip motion is used. The x-axis represents the percentage of motion completion and the motion is averaged over 5 different trials.

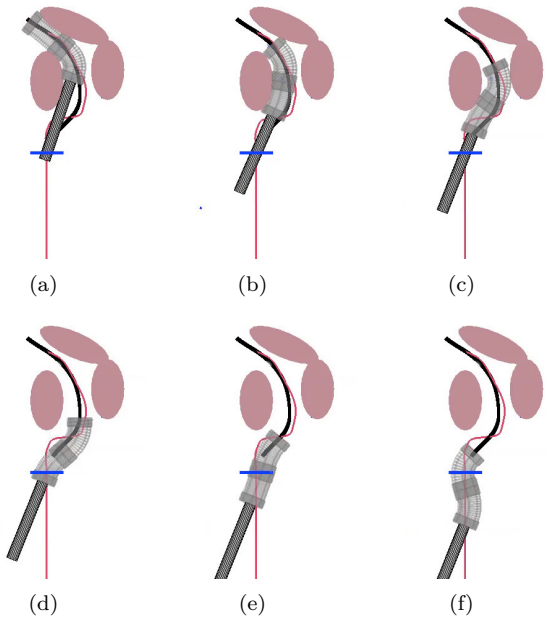

Figure 7: Retraction of the STIFF-FLOP robot from the operation site. The end-effector of the robot is first aligned with the retraction trajectory, represented in black (a). The latter is evaluated as a compromise between tracking precision and curvature minimization. The red trajectory represents the average trajectory that is tracked. The robot is successfully displaced out of the body on a safe trajectory in a completely automated way, that is learned from the movements of the surgeon.

to evaluate the pose from the pressures actuated inside the chambers.

In order to perform the test, the algorithm was embedded inside the ROS architecture of the STIFF-FLOP robot as a secondary task for the inverse kinematics. At each time step, the appropriate command for the null-space motion is evaluated and added to the end-effector command inside the inverse kinematics loop. The whole algorithm takes less than 10 ms to compute the null-space command and to update the statistical model of the free space. The pose of the robot is displayed inside the RViz visualizer from ROS. The visualization is made up of rigid cylinders, simulating the rigid connector between the flexible parts.

The results of a simple insertion task are shown in Fig. 8 and 9.

The robot is moved on one side to avoid a circular obstacle in front of it, inserted inside the environment and then brought back towards the central line. Since the area in front of the robot was not explored by the end-effector, the algorithm considers it as a forbidden zone, and the body is pushed away from it. After the insertion is completed the end effector is moved in the perpendicular direction back and forth, simulating a cutting behaviour. During this phase, the robot keeps the curved shape and avoids the obstacle with its body.

In. Fig. 10 the Gaussians representing the free space learnt by the algorithm are shown. The first big Gaussian represents the initial estimate of the free space around the robot, which is decided by the user.

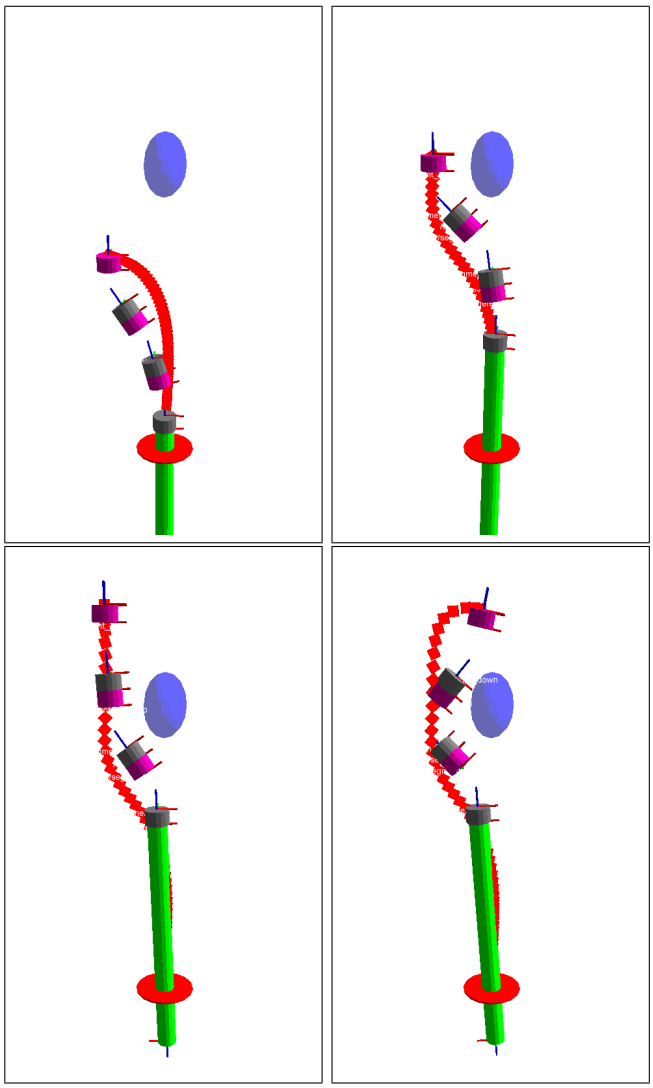

Figure 8: Insertion task while avoiding obstacle. The body of the robot is displaced on the left and kept away from the obstacle when the end effector comes back towards the central line. The blue ellispoid represents an obstacle, while the magenta and grey cylinders are the rigid sections between the flexible modules. The red trajectory represents the best evaluated optimal shape, while the black grid on the top is the target tissue to cut. The robot is not sticking to the trajectory exactly because of the minimal intervention principle. The robot is exploiting the variability in the lower part of the space to reorient the robot.

7 Discussion

The algorithm explained in the current paper is fully operational and generalizes the approach presented in (Bruno et al., 2014). In fact, the previous algorithm relied on multiple demonstrations of the task that needed to be given in advance to show the variability allowed to the body of the robot. Moreover, the current paper formalizes the previous approach by exploiting the LQR approach to generate the null space body movements.

The usefulness of the present algorithm is placed in the possibility of controlling the body of the manipulator without any external intervention from the user. In fact, the surgeon is often not even aware of the motion of the manipulator body. A full teleoperation would be too complex to be performed, both from a technical viewpoint (need for additional cameras) and from the teleoperator viewpoint (too many degrees of freedom to control with only two hands).

The choice of the learning by demonstrations approach allows us to avoid pre-programming task specific obstacle avoidance behaviour inside the robot. The solution is scalable and adapts to very different environments, since the

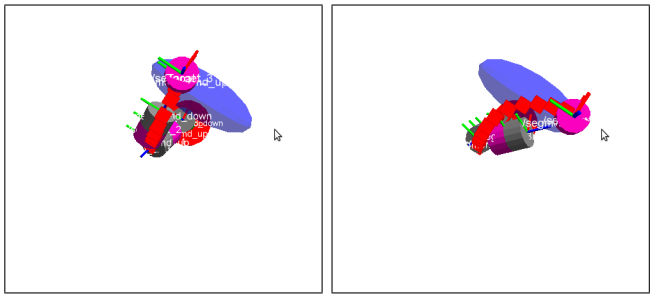

Figure 9: After insertion, the end effector is performing a cutting task behind the obstacle. During all the cutting, the body of the robot still avoids the obstacle while allowing the end effector to be moved freely to perform the task. Notice that the body of the robot remains still while the end effector moves along the cutting trajectory, since no variability was demonstrated to it.

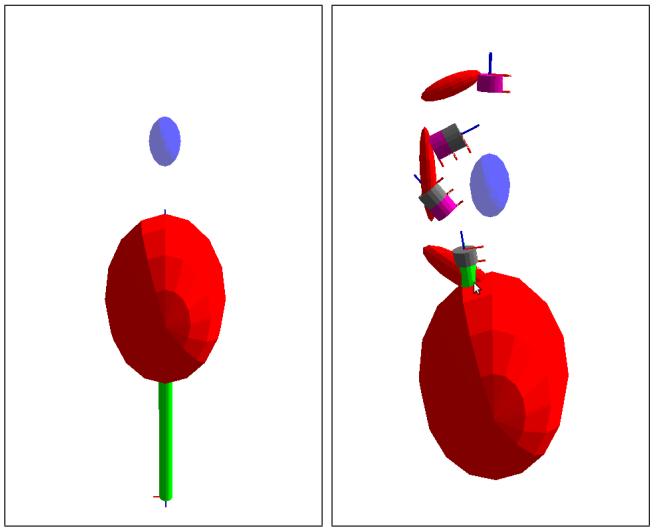

Figure 10: The GMM representing the empty space along the task. Left: the initial variability around the robot (set by the user). Right: the variability at the end of the task.

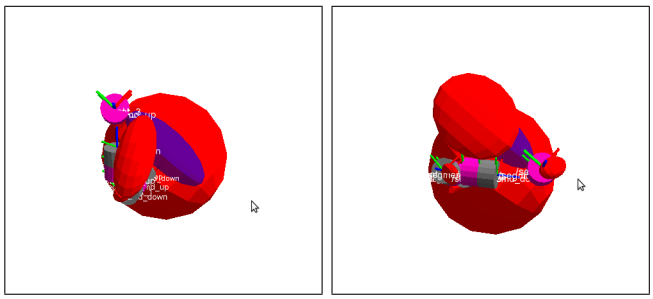

Figure 11: The modification of the GMM before (left) and after (right) the cutting is performed. Notice that the variability allowed in the last part of the task grows, while it remains invariant for the rest of the body.

demonstrations are collected along with the task, without needing any previous information.

An alternative would be to choose some intermediate viapoints defined by the surgeon that the robot is forced to pass during the execution of the task, in order to avoid some proscribed areas defined by the operator during the surgery. Yet, this choice obliges the surgeon to put additional effort in correctly selecting the viapoint positions. In also forces him to follow a single trajectory during the approach phase. Instead, the current approach only requires the surgeon to perform a preliminary exploration of the area, which needs to be performed in any case, as the operational procedure demands (Yang et al., 2012). Furthermore, the safety is increased as well, since an incomplete preliminary exploration will constrain the robot more than it is needed, while a missing viapoint could potentially create a danger.

The advantage of using a statistical model, as opposed to standard planning algorithms, is placed in its greater versatility. In fact, the use of planning techniques that create a network of spheres around the movements of the end-effector would model the free space in a similar manner as the GMM does. This could be exploited by standard planning algorithms to find a correct motion for the body of the robot (see e.g. Quinlan and Khatib (1993)).

Nevertheless, the planning techniques would not provide the encoding of the correlation between the elongation of each section and the desired Cartesian position, forcing us to create an additional null-space controller for the length of the modules. Actually, a GMM can be considered as a statistical version of the sphere network, which is able to encode a greater number of information, together with a statistical interpretation and an automatic organization of the elements in the network.

Moreover, the learning from demonstrations approach avoids the need of coding an explicit controller for the body, since the same algorithm that is used to learn the body motion is also employed within the robot architecture to learn relevant skills from demonstrations. This was demonstrated by re-using movement skills during the retraction phase of the experiment, which were previously acquired during the insertion phase.

In future work, one of the desired extensions would be to make the algorithm able to cope with environmental changes, that could make some areas forbidden to the body of the robot during the surgery. In this case, the presence of a model would ensure a fast intervention, since the latter could be modified during the surgery, by biasing the GMM depending on the input from other sources (cameras, sensors along the body of the robot, contact information, etc.).

8 Conclusion

This paper addresses the problem of controlling the body of a novel flexible surgical robot while the surgeon teleoperates the end-effector of the robot. The envisaged solution is based on the learning from demonstrations paradigm and consists of an algorithm that learns in an online manner the correct behaviour from the motions of the tip that the surgeon naturally performs during the surgical task.

The algorithm exploits the movements of the surgeon to build a statistical model of the areas where the manipulator can freely move without touching the surrounding environment; the model is incrementally built at each time step as the task is progressively executed.

The model is then exploited to implement a positional controller for selected points around the manipulator, that are kept within the admissible space without any input from the surgeon, who is left free to concentrate on the teleoperation of the tip to perform the surgical task.

The algorithm is demonstrated in simulation, by exploiting the inverse kinematics representation of the robot that is used on the real robot. The simulated experiments show how the algorithm can be exploited to fully automate some simple but relevant tasks, such as the safe extraction of the manipulator after surgery.

References

Bruno, D., Calinon, S., and Caldwell, D. G. (2014). Null space redundancy learning for a flexible surgical robot. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2443–2448, Hong Kong, China.

Calinon, S., Bruno, D., and Caldwell, D. G. (2014a). A task-parameterized probabilistic model with minimal intervention control. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3339–3344, Hong Kong, China.

Calinon, S., Bruno, D., Malekzadeh, M. S., Nanayakkara, T., and Caldwell, D. G. (2014b). Human-robot skills transfer interfaces for a flexible surgical robot. *Computer Methods and Programs in Biomedicine*, 116(2):81–96.

Calinon, S., D'halluin, F., Sauser, E. L., Caldwell, D. G., and Billard, A. G. (2010). Learning and reproduction of gestures by imitation: An approach based on hidden Markov model and Gaussian mixture regression. *IEEE Robotics and Automation Magazine*, 17(2):44–54.

Cianchetti, M., Ranzani, T., Gerboni, G., De Falco, I., Laschi, C., and Menciassi, A. (2013). Stiff-flop surgical manipulator: mechanical design and experimental characterization of the single module. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3567–3581.

Flash, T. and Hogan, N. (1985). The coordination of the arm movements: an experimentally confirmed mathematical model. *Neurology*, 5(7):1688–1703.

Gauvain, J.-L. and Lee, C.-H. (1994). Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chians. *IEE Transactions on Speech and Audio Processing*, 2(2):291–298.

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

Jiang, A., Ataollahi, A., Althoefer, K., Dasgupta, P., and Nanayakkara, T. (2012a). A variable stiffness joint by granular jamming. In *ASME Intl Design Engineering Technical Conf. and Computers and Information in Engineering Conf. (IDETC/CIE)*, pages 267–275.

Jiang, A., Xynogalas, G., Dasgupta, P., Althoefer, K., and Nanayakkara, T. (2012b). 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)*, pages 2922–2927.

Kulis, B. and Jordan, M. I. (2012). Revisiting k-means: New algorithms via bayesian nonparametrics. In *Proc. Intl Conf. on Machine Learning (ICML), Edimburgh (UK)*.

Lee, D. and Ott, C. (2011). Incremental kinesthetic teaching of motion primitives using the motion refinement tube. *Autonomous Robots*, 31(2):115–131.

Lobaton, E., Fu, J., Torres, L., and Alterovitz, R. (2013). Continuous shape estimation of continuum robots using x-ray images. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 725–732.

Lyons, L., Webster, R., and Alterovitz, R. (2010). Planning active cannula configurations through tubular anatomy. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2082 – 2087, Anchorage, Alaska.

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In *Proc. Symposium on Mathematical Statistics and Probability*, pages 281–297, Berkeley. University of California Press.

Malekzadeh, M. S., Calinon, S., Bruno, D., and Caldwell, D. G. (2014). Learning by imitation with the STIFF-FLOP surgical robot: A biomimetic approach inspired by octopus movements. *Robotics and Biomimetics, Special Issue on Medical Robotics*, 1(13):1–15.

Nordmann, A., Emmerich, C., Ruether, S., Lemme, A., Wrede, S., and Steil, J. (2012). Teaching nullspace constraints in physical human-robot interaction using reservoir computing. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1868–1875.

Quinlan, S. and Khatib, O. (1993). Elastic bands: Connecting path planning and control. In *Proc. of the Intl Conf. on Robotics and Automation ICRA*, pages 802–807.

Rajiv Ranganatan, A. A. and Mussa-Ivaldi, F. A. (2013). Learning to be lazy, exploiting redundancy in a novel task to minimize movement-related effort. *Journal of Neuroscience*, 33(7):2754–2760.

Reiter, A., Goldman, R. E., Bajo, A., Iliopoulos, K., Simaan, N., and Allen, P. K. (2011). A learning algorithm for visual pose estimation of continuum robots. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2390 – 2396, San Francisco, CA.

Song, M. and Wang, H. (2005). Highly efficient incremental estimation of Gaussian mixture models for online data stream clustering. In *Proc. of SPIE: Intelligent Computing - Theory and Applications III*, volume 5803, pages 174–183.

Sternad, D., Abe, M. O., Hu, X., and Mueller, H. (2011). Neuromotor noise, error tolerance and velocity-dependent costs in skilled performance. *PLoS Computational Biology*, 7(9).

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

Towell, C., Howard, M., and Vijayakumar, S. (2010). Learning nullspace policies. In *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, pages 241–248, Taiwan.

Van Den Berg, J., Miller, S., Duckworth, D., Hu, H., Wan, A., Fu, X., Goldberg, K., and Abbeel, P. (2010). Superhuman performance of surgical tasks by robots using iterative learning from human-guided demonstrations. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2074–2081, Anchorage, Alaska.

Yang, X., Zhu, C., Yang, M., Hu, H., Xia, L., and Pan, K. (2012). Laparoscopic radical resection for rectal cancer. *Translational Gastrointestinal Cancer*, 1(3):255–271.

Zelman, I., Titon, M., Yekutieli, Y., Hanassy, S., Hochner, B., and Flash, T. (2013). Kinematic decomposition and classification of octopus arm movements. *Front Comput Neurosci*, 7:60.

Zhang, Z., Chen, C., Sun, J., and Chan, K. L. (2003). EM algorithms for Gaussian mixtures with split-and-merge operation. *Pattern Recognition*, 36(9):1973–1983.

A Gaussian Mixture Models

The observations $\{\boldsymbol{\xi}_n\}_{n=1}^N$ representing the points of the demonstrations are assumed to be independent realizations of a random vector, that is assumed to be distributed as a linear combination of Normal distributions as

$$\mathcal{P}(\boldsymbol{\xi_n}) = \sum_{k=1}^{K} \pi_k \, \mathcal{N}(\boldsymbol{\xi_n}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \quad \text{with } \mathcal{N}(\boldsymbol{\xi_n}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{(2\pi)^{\frac{D}{2}}|\boldsymbol{\Sigma}_k|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(\boldsymbol{\xi_n}-\boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{\xi_n}-\boldsymbol{\mu}_k)\right].$$

The parameters of a *Gaussian mixture model* (GMM) with K components are thus defined by $\{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$, where π_k is the prior (mixing coefficient), $\boldsymbol{\mu}_k$ is the center, and $\boldsymbol{\Sigma}_k$ is the covariance matrix of the k-th mixture component.

The estimation of mixture parameters can be performed by maximizing the log-likelihood of the above distribution of the given dataset. For the set of observations $\{\boldsymbol{\xi}_n\}_{n=1}^N$, the log-likelihood of the GMM is

$$\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{\xi}) = \sum_{n=1}^{N} \log\left(\sum_{k=1}^{K} \pi_k \, \mathcal{N}(\boldsymbol{\xi}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right). \qquad (12)$$

The maximization of the likelihood leads to an *expectation-maximization* (EM) process iteratively refining the model parameters to converge to a local optimum of the likelihood. These two steps are iteratively applied until a stopping criterion is satisfied. The two steps are described below.

E-step:

$$h_{n,i} \quad = \quad \frac{\pi_i \, \mathcal{N}(\boldsymbol{\xi}_n|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)}{\sum_{k=1}^{K} \pi_k \, \mathcal{N}(\boldsymbol{\xi}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}.$$

M-step:

$$\begin{aligned}\pi_i &= \frac{\sum_{n=1}^{N} h_{n,i}}{N},\\ \boldsymbol{\mu}_i &= \frac{\sum_{n=1}^{N} h_{n,i}\boldsymbol{\xi}_n}{\sum_{n=1}^{N} h_{n,i}},\\ \boldsymbol{\Sigma}_i &= \frac{\sum_{n=1}^{N} h_{n,i}(\boldsymbol{\xi}_n-\boldsymbol{\mu}_i)(\boldsymbol{\xi}_n-\boldsymbol{\mu}_i)^\top}{\sum_{n=1}^{N} h_{n,i}}.\end{aligned}$$

The reproduction of an average movement or skill behavior can be formalized as a statistical regression problem. We demonstrated in previous work that *Gaussian mixture regression* (GMR) offers a simple and elegant solution to handle encoding, recognition, prediction and reproduction in robot learning (Calinon et al., 2010). It provides a probabilistic representation of the movement, where the model can retrieve actions in real-time, within a computation time that is independent of the number of datapoints in the training set.

By defining which variables span for input and output parts (noted respectively by $\mathcal{I}$ and $\mathcal{O}$ superscripts), a block decomposition of the datapoint $\boldsymbol{\xi}_n$, vectors $\boldsymbol{\mu}_i$ and matrices $\boldsymbol{\Sigma}_i$ can be written as

$$\boldsymbol{\xi}_n = \begin{bmatrix}\boldsymbol{\xi}_n^I\\ \boldsymbol{\xi}_n^O\end{bmatrix}, \quad \boldsymbol{\mu}_i = \begin{bmatrix}\boldsymbol{\mu}_i^I\\ \boldsymbol{\mu}_i^O\end{bmatrix}, \quad \boldsymbol{\Sigma}_i = \begin{bmatrix}\boldsymbol{\Sigma}_i^I & \boldsymbol{\Sigma}_i^{IO}\\ \boldsymbol{\Sigma}_i^{OI} & \boldsymbol{\Sigma}_i^O\end{bmatrix}.$$

The GMM thus encodes the joint distribution $\mathcal{P}(\boldsymbol{\xi}^I,\boldsymbol{\xi}^O) \sim \sum_{i=1}^{K}\pi_i\mathcal{N}(\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i)$ of the data $\boldsymbol{\xi}$. At each reproduction step n, $\mathcal{P}(\boldsymbol{\xi}_n^O|\boldsymbol{\xi}_n^I)$ is computed as the conditional distribution

$$\mathcal{P}(\boldsymbol{\xi}_n^O|\boldsymbol{\xi}_n^I) \sim \sum_{i=1}^{K} h_i(\boldsymbol{\xi}_n^I)\,\mathcal{N}\Big(\hat{\boldsymbol{\mu}}_i^O(\boldsymbol{\xi}_n^I), \hat{\boldsymbol{\Sigma}}_i^O\Big), \quad (13)$$

$$\text{with} \quad \hat{\boldsymbol{\mu}}_i^O(\boldsymbol{\xi}_n^I) = \boldsymbol{\mu}_i^O + \boldsymbol{\Sigma}_i^{OI}{\boldsymbol{\Sigma}_i^I}^{-1}(\boldsymbol{\xi}_n^I-\boldsymbol{\mu}_i^I), \quad (14)$$

$$\hat{\boldsymbol{\Sigma}}_i^O = \boldsymbol{\Sigma}_i^O - \boldsymbol{\Sigma}_i^{OI}{\boldsymbol{\Sigma}_i^I}^{-1}\boldsymbol{\Sigma}_i^{IO},$$

$$\text{and} \quad h_i(\boldsymbol{\xi}_n^I) = \frac{\pi_i\mathcal{N}(\boldsymbol{\xi}_n^I|\,\boldsymbol{\mu}_i^I,\boldsymbol{\Sigma}_i^I)}{\sum_k^K \pi_k\mathcal{N}(\boldsymbol{\xi}_n^I|\,\boldsymbol{\mu}_k^I,\boldsymbol{\Sigma}_k^I)}. \quad (15)$$

In the general case, eq. (13) represents a multimodal distribution. In problems where a single output is expected (single peaked distribution), eq. (13) can be approximated by a single normal distribution $\mathcal{N}(\hat{\boldsymbol{\mu}}_n^O, \hat{\boldsymbol{\Sigma}}_n^O)$ with parameters

$$\hat{\boldsymbol{\mu}}_n^O = \sum_i h_i(\boldsymbol{\xi}_n^I)\left[\boldsymbol{\mu}_i^O + \boldsymbol{\Sigma}_i^{OI}{\boldsymbol{\Sigma}_i^I}^{-1}(\boldsymbol{\xi}_n^I-\boldsymbol{\mu}_i^I)\right], \quad (16)$$

$$\hat{\boldsymbol{\Sigma}}_n^O = \sum_{i=1}^{K} h_i(\boldsymbol{\xi}_n^I)\boldsymbol{\Sigma}_i^O + \sum_{i=1}^{K} h_i(\boldsymbol{\xi}_n^I)\boldsymbol{\mu}_i^O(\boldsymbol{\mu}_i^O)^\top - \hat{\boldsymbol{\mu}}_n^O\hat{\boldsymbol{\mu}}_n^{O\top}.$$

Eq. (16) is computed online from the model parameters. The retrieved signal encapsulates variation and correlation information in the form of a probabilistic flow tube, see e.g., (Lee and Ott, 2011).

Algorithm 3 Robot body behaviour control (RBBC).

Require: Number of modules M and precision factor R
Require: Body index $s(i) \in [1, M]$ of the N points to be controlled
Require: Desired length of each module $L = 1.3L_0$
Require: Initial confiquration of the robot ($\boldsymbol{q}$)
Require: Position and orientation of trocar Port ($\boldsymbol{b}$, $\boldsymbol{A}$)
Initialization: $(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k) \leftarrow$ GMM encoding the initial configuration by Alg.2
repeat
  $\theta \leftarrow$ task space motion command by the surgeon
  $\boldsymbol{J} \leftarrow$ jacobian corresponding to current configuration $\boldsymbol{q}$
  $\boldsymbol{P}_i \leftarrow$ position of selected points to be controlled
  $\hat{\boldsymbol{J}}_i \leftarrow$ jacobian of the i-th point to be controlled
  $L_i \leftarrow L(M - s(i))$ desired distance to point i
  $L_{tot} \leftarrow$ total length of manipulator
  $L_{trocar} \leftarrow$ distance of trocar port from base
  $L_{inside} \leftarrow L_{tot} - L_{trocar}$
  if base inside body **then**
    $L_{inside} \leftarrow L_{inside} + |\boldsymbol{b}|$
  end if
  $\boldsymbol{P} \leftarrow$ position of the end-effector
  for each selected point $\boldsymbol{P}_i$ **do**
    Evaluate the Jacobian of $\boldsymbol{P}_i$
    Calculate the distance from trocar
      $\delta_i = L_{inside} - L_i$
    Evaluate the position $\boldsymbol{\mu}^O(\delta_i)$ and variability $\boldsymbol{\Sigma}^O(\delta_i)$
    Calculate the displacement $\boldsymbol{V}_i[\boldsymbol{\mu}_i^O(\delta_i), \boldsymbol{\Sigma}_i^O(\delta_i)]$
    Calculate the corresponding null space velocity
      $\delta\boldsymbol{q}_i = (\boldsymbol{I} - \boldsymbol{J}^\dagger\boldsymbol{J})\hat{\boldsymbol{J}}_i^\dagger\boldsymbol{V}_i$
  end for
  Average null space displacements
    $\delta\boldsymbol{q}_{null} \leftarrow \frac{1}{N}\sum_i \delta\boldsymbol{q}_i$
  Evaluate command for θ : $\delta\boldsymbol{q}_{command} \leftarrow \boldsymbol{J}^\dagger\theta$
  Move the robot : $\boldsymbol{Q} \leftarrow \boldsymbol{Q} + \delta\boldsymbol{q}_{command} + \delta\boldsymbol{q}_{null}$
  Update the GMM distribution :
  $\boldsymbol{\xi}^* \leftarrow [L_{inside}, \boldsymbol{P}^\top]^\top$
  $(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k) \leftarrow$ GMM updated adding $\boldsymbol{\xi}^*$ by using algorithm 1
until end of task