

## **PART III**

# **Control, Kinematics and Navigation**



# 9

---

## Inverse Kinematics Methods for Flexible Arm Control

---

**Anthony Remazeilles, Asier Fernandez Iribar  
and Alfonso Dominguez Garcia**

Tecnalia Research & Innovation, Donostia – San Sebastián, Spain

### **Abstract**

Traditional minimally invasive robots provide to the surgeon an interface for controlling the tip of the endoscopic arm in Cartesian space. We proposed therefore a similar interface for the STIFF-FLOP robot. The direct control of the tip pose was provided by an inverse kinematics component, computing the appropriate STIFF-FLOP robot configuration. Due to the flexibility of the arm modules, we have organized the inverse kinematics into two layers. The first one handles the inverse kinematics in a generic way. It is based on a numerical estimation of the robot. This layer is generic in the sense that it can incorporate any module representation, as long as the module representation provides a forward kinematics mechanism. The second layer concerns the kinematic modeling of the flexible modules, and has to provide forward kinematics functionalities for the upper model. Instead of the standard constant curvature parameters, we are proposing two other representations, one using each module tip position, and the other one directly using the chamber lengths. The flexible modules are connected to a robotic arm through a rigid rod, to extend the operational space of the system. The robotic arm pose is encoded with an adaptation of the spherical coordinate system to ensure that the rod entering the human body respects the single insertion point constraint. By defining a forward kinematics for the rod pose, the external robot end effector is implicitly embedded into the general inverse kinematics scheme, so that the estimation of the flexible modules' configurations and the pose of the robot end-effector are all computed together to follow the motion requests provided by the surgeon.

## 9.1 Introduction

### 9.1.1 On the Inverse Kinematics Problem for Continuum Robots

The design of a user interface for controlling a (piecewise) continuum robot like the STIFF-FLOP arm is challenging, since such robot presents many more control parameters than can be provided through a traditional haptic device such as the one presented in Figure 9.1 (for description of haptic device refer to Chapter 16). Such conventional surgeon interfaces are used for providing the desired six DOF pose of the surgical tool tip, while the actuation space of the robot is much higher. In the case of the STIFF-FLOP robot, each flexible module is controlled with three parameters (pressure in each chamber), so that the flexible arm requires defining  $3n$  parameters,  $n$  being the number of modules used within the arm. If the flexible arm is mounted onto a rigid robot to extend the actuation envelope (Figure 9.2 and section 9.2.4), the positioning of this additional robot needs also to be controlled.

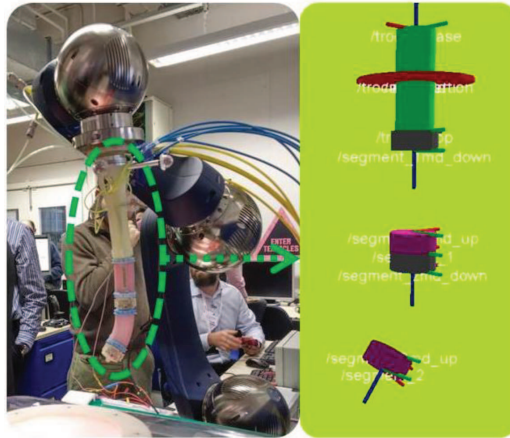
This problem is classical in robotics theory, and is related to *Inverse Kinematics*. As stated in [1] it consists of finding all the geometric parameters of the manipulator given the desired position and orientation of the end-effector. Considering our robot's specificities, the work of Webster et al. is particularly relevant since it provides key kinematics models for piecewise constant curvature continuum robots [2]. Under the piecewise constant curvature assumption, the modeling of such robotic structures can be seen



**Figure 9.1** Example of a haptic device used to receive the motion request for the robotic tool (Omega 7 from Force Dimension<sup>1</sup>). For description refer to Chapter 16.

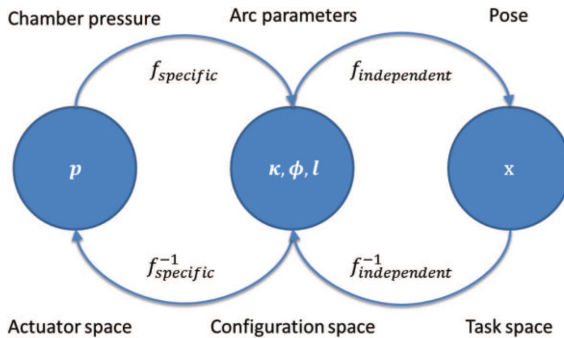
---

<sup>1</sup><http://www.forcedimension.com>



**Figure 9.2** Real STIFF-FLOP system (left) vs. the simulated version used in this chapter (right). A rigid section (green on the right) is connecting the set of flexible modules to the SCHUNK arm. The connection with the first module is what we call the base in this chapter. On the right picture, only the rigid components are displayed. All modules are started and finished by a rigid section (respectively gray and purple). Once the flexible modules are inserted into the body, the motion of the SCHUNK arm must maintain a unique insertion location. In simulations involving the motion of the base, the red disc (top right) represents that constraint.

as a composition of robot-specific mapping and robot-independent mapping (Figure 9.3). The forward kinematics consists of the operations described on the upper side of the figure (going from the actuator space to the task space), while the inverse kinematics focuses on the lower operations (going from the task space to the actuator space).



**Figure 9.3** Spaces and mappings of constant curvature robots, as described in [2].

The use of arc parameters in the configuration space ( $\kappa, \phi, l$ , respectively for the curvature, the angle of the plane containing the arc, and the arc length) to represent the state of a continuum flexible module respecting the constant curvature model makes sense. Nevertheless, the constant curvature model may lead to some kinematic singularities, e.g., when a flexible module is purely extended so that its curvature  $\kappa$  is equal to 0. Furthermore, it does not allow taking into account the potential deviations from the theoretical model, due, for example, to the external forces acting on each module. In the latter case, a Jacobian formulation in closed form of the independent mapping, as proposed in [2], may not be easily obtained.

Following the suggestions of our colleagues in [3], we propose to relax the central role of the constant curvature parameter in modeling, and propose a generic model of the configuration space. We combine this with a numerical computation of the Jacobian. The advantage of using a Jacobian estimated numerically is that it allows considering any configuration space that provides a forward kinematics mapping. This approach is also used to deduce, simultaneous to the estimation of the appropriate configuration of the flexible modules, the motion of the robotic arm holding the STIFF-FLOP flexible modules, while respecting the single insertion point constraint that is described in the following section.

### **9.1.2 Single Insertion Point Constraint in Minimally Invasive Surgery**

In the context of robotic surgery and minimally invasive surgery (MIS), the main purpose of using a tool composed of flexible modules is to permit accessing spaces that cannot be directly reached by a rigid structure, limiting thus the multiplication of insertion points and the number of needed incisions into the human body. Nevertheless, it is still preferable to mount the flexible modules onto a standard robotic arm to augment the reachable space, and only employ the bending properties of the modules when complex displacements are required. Naturally, the combination of the robotic arm with the flexible modules increases the number of actuation parameters, and an automatic control of that robotic arm is required to maintain the classical interface used by surgeons.

Any robotic system involved in MIS must respect the single insertion constraint. The instrument held by the robot is inserted into the human body through a trocar at the incision point, and the trocar position needs to be maintained throughout the surgery. This constraint can be solved by

the design of dedicated mechanical structures that inherently respect the remote center of motion, such as the well-known Da Vinci [4], or endoscope maintainers directly placed onto the body, such as [5]. Another approach consists in equipping the robot with passive joints at the wrist so that the instrument naturally rotates around the fulcrum point [6, 7], but backlash may appear in some configurations leading to a lack of control of the instrument motion.

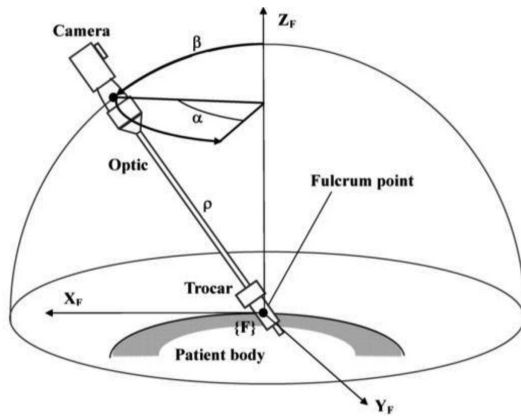
The use of more regular six-DOF robots can be seen as a more versatile solution involving a light robotic system that facilitates its displacement during surgery (in particular when being moved from a trocar to another [8]). The insertion point constraint is ensured by dedicated controllers, leading to a programmable remote center of motion. In [8], the trocar constraint is modeled as a variable point along a given robot link, and is considered as an additional joint added to the arm configuration space ones during the inverse kinematics process. In [9], the task space is extended with the trocar position to produce movements that restrict its motion. In [10], a force sensor is placed at the end effector of the arm to adjust the lateral motion of the arm for limiting the forces applied at the trocar site. Other works based on visual servoing directly adjust the interaction matrix linking the motions of a camera to the image point motion for taking into account the trocar constraint that reduce the displacement of the endoscope [11, 12].

In the context of programmable remote center of motion, the modeling of the trocar constraint with spherical coordinates seems to be particularly appropriate since, *per se*, spherical coordinates can only describe directions going through the origin (which is placed at the trocar frame, as illustrated in Figure 9.4) [10, 13, 14].

We propose in this chapter to embed the robotic arm in the inverse kinematics framework by inserting a component defining the location of the rigid connector between the end effector of the arm and the first flexible module. The pose of the rigid connector is defined with spherical coordinates in which the origin frame is placed at the insertion point of the trocar.

### 9.1.3 Contributions Presented

The present chapter presents how we propose to simultaneously estimate the appropriate configuration of the flexible modules together with the needed robotic arm end-effector pose for reaching a desired tip location provided by the surgeon. The next section describes the generic inverse kinematics framework that is used. It is generic in the sense that it is independent of the



**Figure 9.4** Spherical coordinates used for expressing the pose of an endoscope with respect to the insertion point, as proposed in [13].

configuration space selected for representing the state of the flexible modules (several configuration spaces are considered in section 9.2.3), and it can be applied for any number of flexible modules. Furthermore, it also enables to consider the external robotic arm onto which the flexible modules are mounted, and to deduce the appropriate pose of this arm end-effector while ensuring the trocar constraint. Section 9.2.4 will show how the traditional spherical coordinates can be adjusted to better fit with the surgical context. We will then show in section 9.2.5 that the redundancy of our global system can be used to consider secondary tasks improving the behavior of the flexible modules in the patient body. Finally several simulations will be detailed in section 9.3 to demonstrate the validity of the contributions proposed.

## 9.2 Inverse Kinematics Framework

### 9.2.1 General Framework

We consider that we know the current pose of the STIFF-FLOP tip with respect to a world frame  $W$ . We note it:

$${}^W M_t = \begin{bmatrix} {}^W R_t & {}^W t_t \\ \mathbf{0} & \mathbf{1} \end{bmatrix}, \quad (9.1)$$

and we consider that a desired tip pose is defined, as  ${}^W M_t^*$ . In the spirit of the formalization proposed in [3], the control of the complete STIFF-FLOP



system relies on the definition of a task function  $e$  representing the error between the current tip pose and the desired one. The task error is defined as:

$$\mathbf{e} = [\Delta_t, \theta\mathbf{u}]^T, \quad (9.2)$$

where the first three entries  $\Delta_t$  are related to the difference between the current tool tip position and the desired one:

$$\Delta_t = W_t t_t - W_t t_t^*, \quad (9.3)$$

and  $\theta\mathbf{u}$  is the axis-angle representation of the orientation difference between the current tip frame and the desired one. In [3], the orientation error was only considering the  $x$  and  $y$  components of the rotation vector. To be more general, we extended the error model to contain as well the rotation around the  $z$ -axis, so that the error model represents the complete pose of the tip.

The task function variation can be related to the system parameterization  $\mathbf{q}$ :

$$\frac{d\mathbf{e}}{dt} = \frac{\partial \mathbf{e}}{\partial \mathbf{q}} \frac{d\mathbf{q}}{dt} = \mathbf{J} \frac{d\mathbf{q}}{dt}, \quad (9.4)$$

where  $\mathbf{J}$  is the Jacobian that links the evolution of the task function to the variation of the variables contained in  $\mathbf{q}$ , so that:

$$\dot{\mathbf{q}} = \mathbf{J}^+ \dot{\mathbf{e}} \quad (9.5)$$

The usual models used for the task error evolution  $\dot{\mathbf{e}}$  along time are either affine or exponential. In an affine model  $\dot{\mathbf{e}} = -\lambda\mathbf{e}$ , while in a model with an exponential decrease  $\dot{\mathbf{e}} = -\lambda\mathbf{e}$ .

In the context of the STIFF-FLOP project, we proposed to compute the Jacobian numerically. The advantage of such an approach is that we can seamlessly investigate new module models, by adjusting accordingly the related parameters in  $\mathbf{q}$ , and by providing the related forward kinematics for each piecewise component. The numerical estimation of the Jacobian is based on its structure:

$$\mathbf{J} = \begin{bmatrix} \frac{\partial \mathbf{e}}{\partial q_0} & \dots & \frac{\partial \mathbf{e}}{\partial q_n} \end{bmatrix}, \quad (9.6)$$

So that each column of the Jacobian can be estimated in the following way:

$$\frac{\partial \mathbf{e}}{\partial q_i} = \frac{\mathbf{e}(\mathbf{q} + \delta \mathbf{q}_i) - \mathbf{e}(\mathbf{q})}{\delta} \quad (9.7)$$

### 9.2.2 Application to the STIFF-FLOP Structure

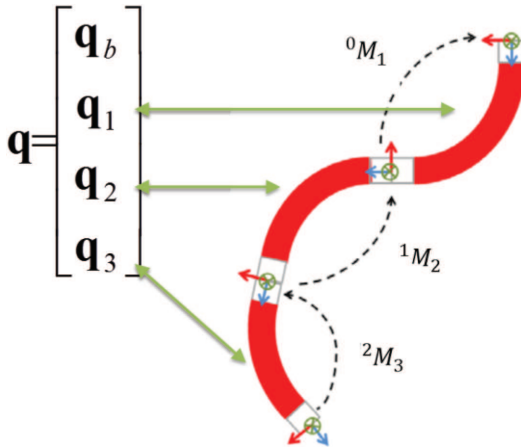
The generic formulation of the structure configuration  $\mathbf{q}$  is defined as a stacking of the configuration of each sub-component constituting the STIFF-FLOP robot: a configuration  $\mathbf{q}_b$  of the base of the flexible modules (the position of which is adjusted using the standard robotic arm holders), and  $n$  sub-configurations specification for each of the  $n$  flexible modules considered (three are presented in Figure 9.5, but the principle is generic).

In the above example, the pose of the tip of the STIFF-FLOP structure can be obtained by the composition of the pose of each successive component. Following the notation used in the previous section, we can write:

$${}^w M_t = {}^w M_b \cdot {}^b M_1 \cdot {}^1 M_2 \cdot {}^2 M_3 \tag{9.8}$$

The first transform is the definition of the pose of the base of the flexible modules, handled with the standard robotic arm. The following transforms are obtained by applying the forward kinematics mapping from the related module configuration  $\mathbf{q}_i$  (and including the connecting rigid sections that are skipped for notation simplicities).

As previously stated, the choice of the parameterization format of each module and of the base is transparent for the inverse kinematics model based on a numerical estimation of the Jacobian. It is only necessary to have a



**Figure 9.5** Relation between the parameterization of the STIFF-FLOP structure and the tip pose.  $\mathbf{q}_b$  refers to the configuration of the base of the flexible structure, which is controlled through the robotic holder not depicted here. Each module is equipped with two rigid connections to allow inter-module attachments.

forward kinematic mapping from this configuration space, to provide by composition the expression of the tip location with respect to a reference world frame. In particular, the expression of the base location  $\mathbf{q}_b$  is different from the one used for the flexible modules  $\mathbf{q}_i$ , since they correspond to components which are totally different. But they can be stacked in a common vector to produce the global configuration space of the STIFF-FLOP robot.

### 9.2.3 Configuration Space of the Flexible Modules

In [2], and as illustrated in Figure 9.3, the configuration space chosen for the flexible modules is directly the constant curvature parameterization, so that each module will be represented by the feature  $\mathbf{q}_i = [\kappa_i, \phi_i, l_i]^\top$ . Such parameterization suffers from a representation singularity when a module is purely extended: the curvature is null, and the plane angle  $\phi$  can have any value.

In [3], it is proposed to use the position of the tip of each flexible module with respect to its base (noted  $\mathbf{Q}_i$ ). This means that the parameterization  $\mathbf{q}$  is defined as (omitting the base pose):

$$\mathbf{q} = [\mathbf{Q}_0, \dots, \mathbf{Q}_{k-1}]^\top. \quad (9.9)$$

The forward kinematics of a module from such configuration requires the computation of the module orientation  $\mathbf{R}_i$ , which is under the constant curvature assumption [3], assuming  $\mathbf{Q}_i = [x_i, y_i, z_i]^\top$ :

$$\mathbf{R}_i = \frac{\mathbf{1}}{\mathbf{Q}_i^\top \mathbf{Q}_i} \begin{bmatrix} -x^2 + y^2 + z^2 & -2xy & 2xz \\ -2xy & x^2 + z^2 - y^2 & 2yz \\ -2xz & -2yz & z^2 - x^2 - y^2 \end{bmatrix} \quad (9.10)$$

As stated, other models can be considered in this generic inverse kinematics framework. For instance, instead of the tip position, one can use the length of the pressured chambers, *i.e.*,  $\mathbf{q}_i = [l_1, l_2, l_3]^\top$ . In that case, the forward kinematics requires computing the constant curvature parameters of each module (noting here the constant curvature with the bending angle  $\alpha$ , the orientation angle  $\beta$  and the chamber length  $L$  as in [15]):

$$\begin{aligned} L &= \frac{1}{3} (l_1 + l_2 + l_3) \\ \alpha &= a \tan 2 \left( \sqrt{3} (l_3 - l_2), l_2 + l_3 - 2l_1 \right) \\ \beta &= \frac{2\sqrt{l_1^2 + l_2^2 + l_3^2 - l_1 l_2 - l_1 l_3 - l_2 l_3}}{3r} \end{aligned}$$

with  $r$  being the radial distance of the chamber from the module center. From the constant curvature model, the pose of the tip of a module with respect to its base is:

$${}^b\mathbf{t}_t = \frac{L}{\beta} \begin{bmatrix} \cos \alpha (1 - \cos \beta) \\ \sin \alpha (1 - \cos \beta) \\ \cos \beta \end{bmatrix}. \quad (9.11)$$

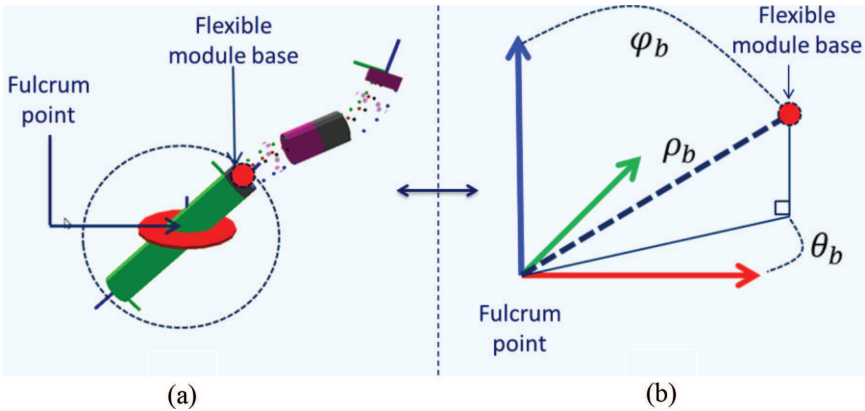
And the orientation is obtained through:

$${}^b\mathbf{R}_t = \begin{bmatrix} \cos^2 \alpha (\cos \beta - 1) + 1 & \sin \alpha \cos \alpha (\cos \beta - 1) & \cos \alpha \sin \beta \\ \sin \alpha \cos \alpha (\cos \beta - 1) & \cos^2 \alpha (1 - \cos \beta) + \cos \beta & \sin \alpha \sin \beta \\ -\cos \alpha \sin \beta & -\sin \alpha \sin \beta & \cos \beta \end{bmatrix} \quad (9.12)$$

The two previous module models rely on the constant curvature assumption. Even though it is not demonstrated here, it is possible to use models relaxing that hypothesis, as it is done in the work of [16]. In this beam theory-based model, the forces measured at each module junction are taken into account to better estimate the deformation of each flexible module (which in this case does not follow the constant curvature hypothesis). With this model, the configuration space of  $\mathbf{q}$  would be equivalent to the actuator space, i.e., the pressures being applied in the chamber. The forward kinematic model proposed in [16] is compatible with our numerical estimation of the Jacobian. The only difference is that forward kinematics is computed for all flexible modules simultaneously, while in the two previous models introduced here (based on module tip poses and based on chamber lengths), forward kinematics is computed per module independently and then composed.

## 9.2.4 STIFF-FLOP Base Motion with Single Insertion Point Constraint

As already stated in section 9.1.2, mounting the flexible manipulator onto a standard robotic arm enables extending the reachable workspace, while focusing the use of the bending capabilities of the flexible manipulator to areas not reachable to rigid structures with a linear motion. In the context of the STIFF-FLOP project, the flexible manipulator was fixed with a rigid rod at the end-effector of a SCHUNK LWA (as illustrated in Figure 9.2). In the rest of the section, we define the position of the SCHUNK arm by the location of the base of the first flexible module (i.e., just after the green rigid section presented in Figures 9.2 and 9.6). We assume that the standard robotic arm



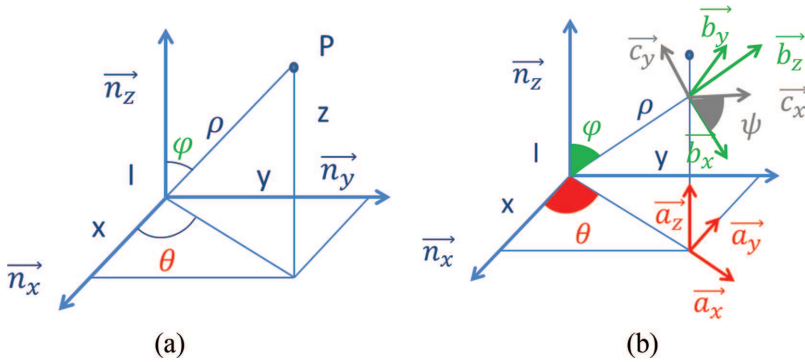
**Figure 9.6** Illustration of the representation of the flexible STIFF-FLOP arm fixed onto a movable structure. (a) The flexible modules are mounted onto a rigid rod (green cylinder), which must respect a fulcrum point (at the center of the red disc). The red dot indicates the tip of the rigid rod that is expressed using spherical coordinates. (b) Spherical coordinate model used to express the pose of the rigid rod tip with respect to an origin placed at the fulcrum point. The red-green-blue vectors represent the reference frame.

is equipped with inverse kinematics means to directly control the pose of the end-effector.

In Figure 9.6, the green cylinder emulates the rigid component onto which the base of the first flexible module is mounted. Again, this rigid component is mounted itself at the end effector of a robotic arm (which would be placed at the bottom left end of the green cylinder on Figure 9.6) providing motion capabilities to the base of the STIFF-FLOP arm. The red disc emulates the insertion point (i.e., the trocar port). Any generated motion of the global system should go through that point. Using spherical coordinates for the robot base, the fulcrum constraint can easily be computed and ensured.

It is straightforward to take the base motion into account within the inverse kinematic model previously defined. Indeed, one can define a specific feature,  $\mathbf{B}$ , gathering the needed parameters to define the base pose with respect to a world frame. If the base would be totally free of motion (the so-called *free-flying base*), then the base model could be chosen to be a vector of six components:

$$\mathbf{q}_b = \begin{bmatrix} \mathbf{w}_{t_b} \\ \text{euler}({}^W\mathbf{R}_b) \end{bmatrix}, \quad (9.13)$$



**Figure 9.7** Spherical coordinates (a) parameters  $\rho$ ,  $\theta$ ,  $\varphi$  stand respectively for the radial distance, the azimuthal angle and the polar angle. (b) Frames generated by a spherical parameterization. Green: standard frame derived from the coordinates. Gray: adding a fourth dimension to model the orientation along the  $z$ -axis.

where the Euler function provides the three Euler angles related to a rotation matrix. In that case, the whole joint vector is defined as:

$$\mathbf{q} = \left[ \mathbf{q}_b^\top, \mathbf{Q}_0, \dots, \mathbf{Q}_{k-1} \right]^\top, \quad (9.14)$$

and the inverse kinematics is handled following the same methodology as previously defined.

Nevertheless, the robotic arm motion must be constrained to make sure that the single point insertion constraint is respected. As illustrated in Figures 9.6 and 9.7, this insertion point denoted by I (located on the abdominal wall) acts as a pivot point or fulcrum, and spherical coordinates can be used to specify the position of the rigid rod P with respect to the reference frame located at I.

The use of spherical coordinates  $[\rho, \theta, \varphi]^\top$  is a convenient way to restrict the possible rotations of a frame attached to P to the ones that can be generated by the pivot point I. As can be seen in Figure 9.7(b), the two angles  $\theta$  and  $\varphi$  define the orientation of the green frame with respect to the blue reference one. Note that the green frame has its  $z$ -axis aligned with the vector  $\vec{IP}$ .

The related frame orientation can be deduced from the spherical coordinates:

$${}^I\mathbf{R}_B = \begin{bmatrix} \cos \theta \cos \varphi & -\sin \theta & \cos \theta \sin \varphi \\ \sin \theta \cos \varphi & \cos \theta & \sin \theta \sin \varphi \\ -\sin \varphi & 0 & \cos \varphi \end{bmatrix} \quad (9.15)$$

Considering that the rotation matrix column refers to the expression of  $\vec{b}_z$  in the reference frame  $\mathbf{I}$ , the position of point  $\mathbf{P}$  in frame  $\mathbf{I}$  is directly deduced as:

$${}^I \mathbf{t}_B = \rho \vec{b}_z = \rho \begin{bmatrix} \cos \theta \sin \varphi \\ \cos \theta \cos \varphi \\ \sin \theta \end{bmatrix}. \quad (9.16)$$

Thus, the configuration of the base can be defined by the spherical coordinates (i.e.,  $\mathbf{q}_b = [\rho, \theta, \varphi]^\top$ ), and the two previous relations correspond to the forward kinematics relations needed for the Jacobian estimation.

In Figure 9.7(b), another dimension is added to the spherical coordinates to model the rotation along the  $z$ -axis (gray frame). In that case, the orientation of the obtained frame can be defined as:

$$\begin{aligned} {}^I \mathbf{R}_C &= {}^I \mathbf{R}_B \mathbf{R}_\psi \\ &= \begin{bmatrix} \cos \theta \cos \varphi & -\sin \theta & \cos \theta \sin \varphi \\ \sin \theta \cos \varphi & \cos \theta & \sin \theta \sin \varphi \\ -\sin \varphi & 0 & \cos \varphi \end{bmatrix} \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (9.17)$$

$${}^I \mathbf{R}_C = \begin{bmatrix} c\theta c\varphi c\psi - s\theta s\psi & -c\theta c\varphi s\psi & -s\theta c\psi & c\theta s\varphi \\ s\theta c\varphi c\psi + c\theta s\psi & -s\theta c\varphi s\psi + c\theta c\psi & s\theta s\psi & s\theta s\varphi \\ -s\varphi c\psi & -s\varphi s\psi & c\varphi & \end{bmatrix}$$

with  $c$ ,  $s$  standing respectively for *cos* and *sin* operators.

As expected, the position of the point  $\mathbf{P}$  (third column) in the reference frame remains unchanged.

Once more, this model,  $\mathbf{q}_b = [\rho, \theta, \varphi, \psi]^\top$ , can be included within our inverse kinematics, considering the forward kinematics provided by the relations previously introduced.

In [17], we have proposed an adjustment of the traditional spherical coordinates to get a better behavior in the context of the trocar constraint within MIS. Looking at the rotation described in Equation (9.17), we can note that the orientation induced by the extended spherical coordinates is similar to Euler angles in the ZYZ configuration of the rotations. It turns out that the system cannot thus directly handle pure rotations around the  $x$ -axis, but has to combine rotations around the  $z$ - and  $y$ -axes to produce them. The proposed adjustment consists of changing the reference angle to switch to a XYZ model instead. As stated in [17], it also enables the elimination of a representation singularity occurring with the traditional model when the green rigid component is aligned with the reference  $z$ -axis of the insertion frame. Readers may refer to [17] for further details on this model adjustment.

### 9.2.5 Secondary Tasks through Redundancy

Redundant robotic systems permit considering additional tasks while satisfying a main task. Our robotic system, controlled by  $4+3n$  parameters (considering the mobile base, and  $n$  flexible modules), operates in a 6-dimensional task space and is, thus, highly redundant. A secondary task can thus be applied in the null space of the first task. This null space can be derived from the Jacobian matrix:

$$\mathbf{P}_e = \mathbf{I}_6 - \mathbf{J}^+ \mathbf{J}. \quad (9.18)$$

Any secondary task projected onto this null space can thus be taken into account without affecting the main task completion. The term “secondary” is used considering that the main task remains the tip positioning, and the second one is only applied onto the null-space left by the primary task, i.e., the secondary task is applied only as long as it does not affect the main one.

The null space can be used to try maintaining the modules around their mean length (to avoid too large an extension in a single module) for instance, or to limit the contact with the environment which could be sensed, for example, by tactile sensors.

#### 9.2.5.1 Control of the chamber lengths

The redundancy is frequently used for joint limit avoidance. In our case, the joint limits are related to the rest of the length of the module (since we cannot reduce the chamber length any further) and the maximum chamber extension. We can then define a secondary task to maintain each module, as much as possible, at its mean size,  $\bar{L}$ .  $\bar{L}$  can be defined for example as  $\bar{L} = L_0 \left(1 + \frac{\alpha}{2}\right)$  where  $L_0$  is the length when the module is at rest and  $\alpha$  is an elongation factor. We can thus define another task function  $W_2$  relating the distance of the modules to their mean size:

$$W_2 = \frac{\sum |L_i - \bar{L}|}{n\bar{L}}. \quad (9.19)$$

Employing a gradient projection approach, we compute the gradient of the original task function  $W_2$ ,

$$\nabla_{\mathbf{q}} W_2 = \frac{\partial W_2}{\partial \mathbf{q}_i} = \frac{\mathbf{W}_2(\mathbf{q} + \delta \mathbf{q}_i) - \mathbf{W}_2(\mathbf{q})}{\delta}. \quad (9.20)$$

and we project the gradient onto the null space of the primary task, so that:

$$\dot{\mathbf{q}} = \mathbf{J}^+ \dot{\mathbf{e}} + \mathbf{P}_e \nabla_{\mathbf{q}} W_2, \quad (9.21)$$



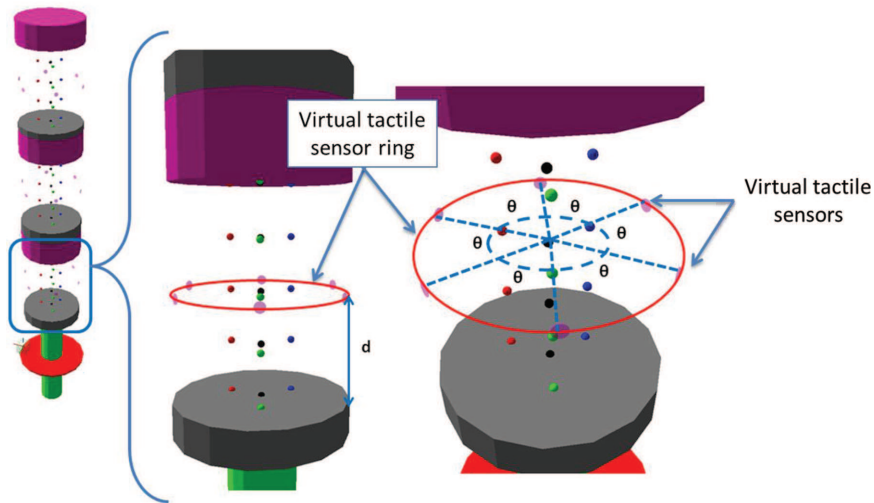
which ensures that the resolution of the second task will be obtained as long as it does not interfere with the primary task.

### 9.2.5.2 Control of the interaction with the environment

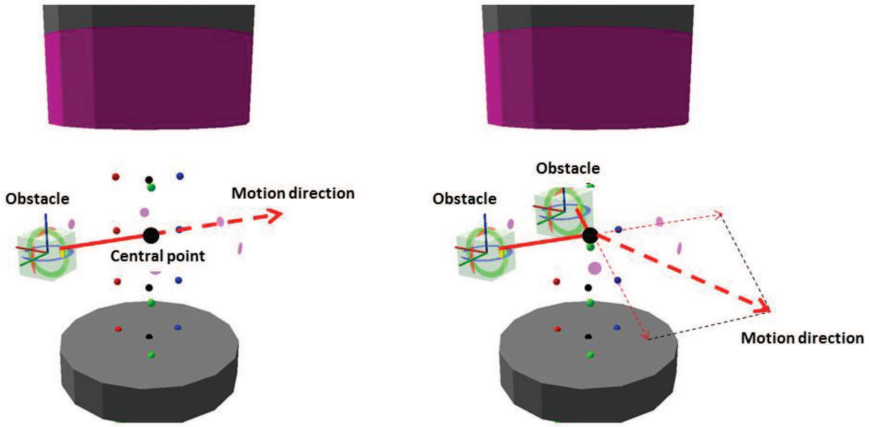
The robot's redundancy can also be used to handle and resolve undesirable interactions with the environment. A control feature optimizes the overall configuration based on input from tactile sensors mounted along the arm or through distributed force/torque sensors. Obstacle avoidance is then activated when the sensed interaction with surrounding soft tissue reaches a defined threshold – to limit the magnitude of the physical interaction with sensitive organs, for instance.

In order to conduct experiments for such a potential extension, we equipped the virtual STIFF-FLOP arm model with distributed tactile sensors as illustrated in Figure 9.8.

When contact is detected, the strategy we propose here is to request a motion of the module in the opposite direction with respect to the central line. The direction of this motion depends thus on the sensors which have been activated (as illustrated in Figure 9.9). Note that such repulsion strategy



**Figure 9.8** Virtual tactile sensors placed along the STIFF-FLOP arm. The tactile sensors are emulated at the locations depicted in purple (middle figure). The right figure shows sensors being equally distributed along the outer layer of the flexible structure, operating like a tactile sensor ring.



**Figure 9.9** Motion request generated after contact detection by a single (left) or by multiple sensors (right).

is a proposition, and could be changed with any other function. We could, for example, accept contacts if the contact force goes beyond a predefined threshold. Any reaction strategy can be considered as long as it can be formulated as a function.

The design of a secondary task related to obstacle avoidance works as follows: Let us consider the positioning task as the priority task, and the obstacle avoidance as a secondary task. This can be formulated as:

$$\dot{\mathbf{q}} = \mathbf{J}^+ \dot{\mathbf{e}} + (\mathbf{J}_o \mathbf{P}_e)^+ (\mathbf{q}_o - \mathbf{J}_o (\mathbf{J}^+ \dot{\mathbf{e}})) , \quad (9.22)$$

where  $\{\mathbf{q}_o, \mathbf{J}_o\}$  refers to the secondary task obstacle avoidance. The first component is set as the desired velocity to move away from the contact (as illustrated in Figure 9.9), and  $\mathbf{J}_o = \frac{\partial \mathbf{X}_o}{\partial \mathbf{q}}$  is the Jacobian relating the motion of the joint variables to the motion of the central point of the tactile ring. In the global equation previously described, the component  $\mathbf{J}_o (\mathbf{J}^+ \dot{\mathbf{e}})$  permits taking into account the motion induced by the primary task that is likely to interfere with the secondary task when combined. Therefore, since the motion direction is defined by the sensor sensing the contact (as explained in Figure 9.9), the insertion of the task for obstacle avoidance only requires defining an appropriate magnitude for the motion generated, and to be able to compute the Jacobian all along the virtual central length (assuming the tactile ring can be placed anywhere along each STIFF-FLOP module).

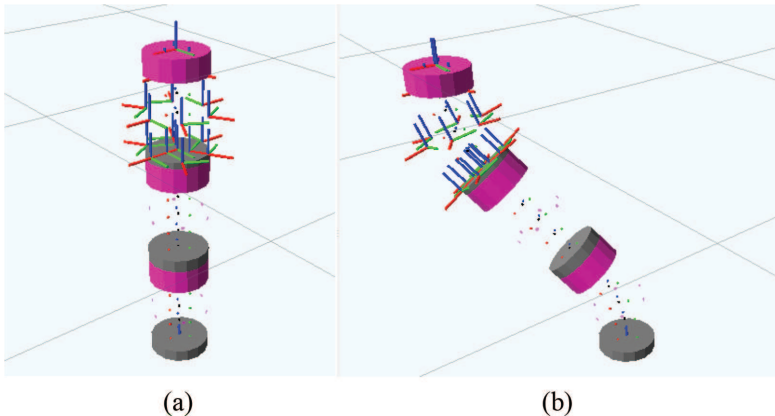
### 9.3 Inverse Kinematic Experimentations

This section demonstrates with simulations the inverse kinematics frameworks previously defined. In all experiments, the inverse kinematics model is requested to produce a bigger motion than what should be effectively requested with the real system. When controlling the real system, the requested motion is always a small displacement (in position and/or orientation) with respect to the current pose, and therefore the underlying motion request is likely to be quite small between successive motion commands. Nevertheless, for the purpose of this chapter, it is more convenient to show the inverse kinematics behavior considering larger motions.

#### 9.3.1 Fixed Base, Various Module Representation

In these first experiments, we consider that the base is not active, i.e., the base of the first module remains fixed, and only the flexible modules are controlled to reach the target pose. The STIFF-FLOP structure considered is composed of three flexible modules.

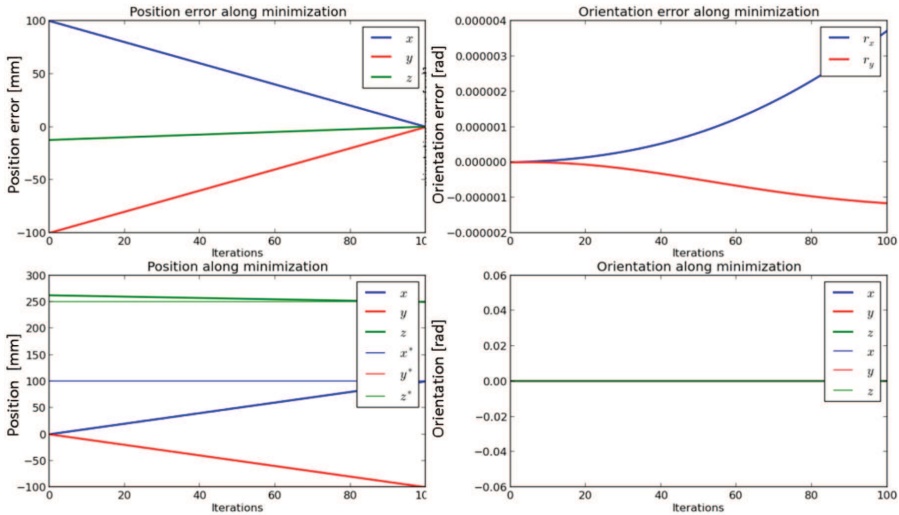
The first illustrated experiment is a pure translational motion along the  $x$ - and  $y$ -axes. Figure 9.10 presents the initial configuration of the arm (left), and the desired final pose (right). In order to compute the appropriate



**Figure 9.10** Experiment 1: Motion experiments with three modules and a fixed base, keeping a fixed orientation of the tip. (a) Initial configuration of the system. (b) Desired final configuration. The pressurized chambers are visualized through small dots between the rigid sections (gray: bottom rigid section of a module, purple: upper rigid section of a flexible module).

configuration, the information provided to the inverse kinematics is the pose of the tip frame as observed in the right figure. In the figures, the frame placed at the middle of the top pink disc corresponds to the tip location of the arm. The desired motion is a pure translation, being  $[0.1, -0.1, 0.25]^T$  with respect to the reference world frame.

In the first experimentation presented in Figure 9.11, the configuration of each module is encoded in  $q$  with the coordinates of each module tip position. Figure 9.11 presents details of the related minimization process by showing the evolution of the pose (position and orientation) error in the upper row, and the evolution of the tip pose along the iteration in the lower row. In all experiments (unless stated otherwise), the convergence is set to get an affine evolution of the error along the iterative minimization process, which is clearly observed with respect to the position, in the two left figures. To do so, the initial pose error is computed and, at each iteration, the algorithm is requested to compensate for a part of this error. In this experiment, the system was requested to converge within 100 iterations. Note that the completion of the requested motion can only be obtained within the pre-specified number of iterations if the Jacobian is well-conditioned.

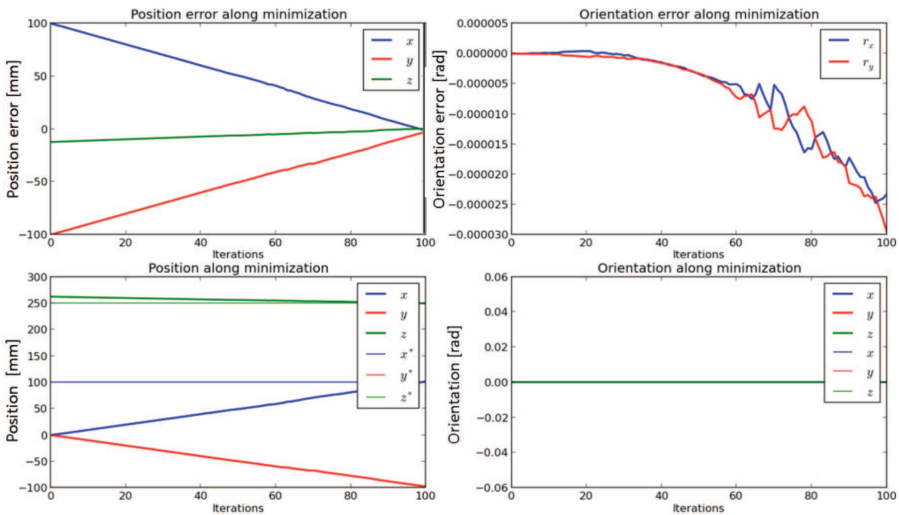


**Figure 9.11** Inverse kinematic output: Upper graphs show the tip position and orientation error along the minimization process. Lower graphs present the tip position and orientation during the minimization. In the bottom graphs, the bold lines show the current values at each iteration, and the horizontal light lines show the desired values (only observable here on the position graph since the initial tip orientation is to be maintained in this example).

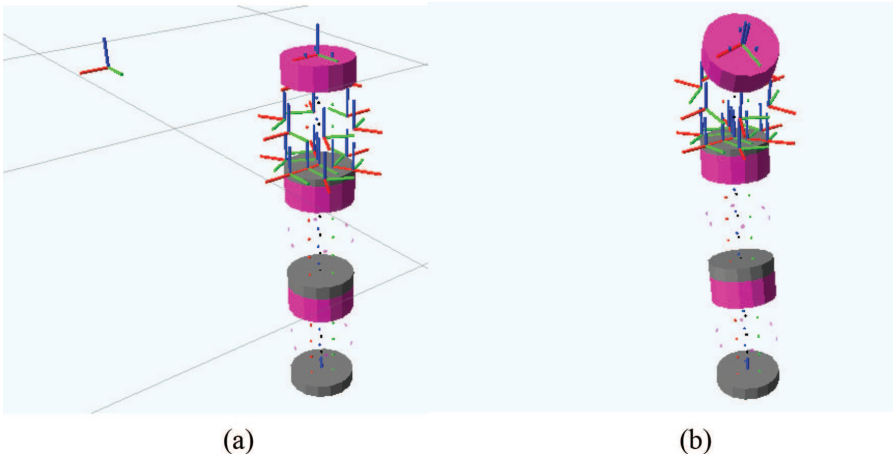
The evolution of the orientation error does not seem to respect the desired affine decrease along time. Looking at the magnitude of the vertical axis, we can see that the error depicted is negligible ( $4 \times 10^{-6}$  rad, or around 0.2 millidegree), and hence can be considered to be null.

While in the experiment of Figure 9.11, the configuration of each flexible module was encoded with the module tip position, the feature used in the experiment presented in Figure 9.12 is the length of each chamber. As it can be observed, despite the use of a different feature, the minimization process is not affected, and the same pose can be obtained as well. Once more, the orientation error is sufficiently small as to be neglected.

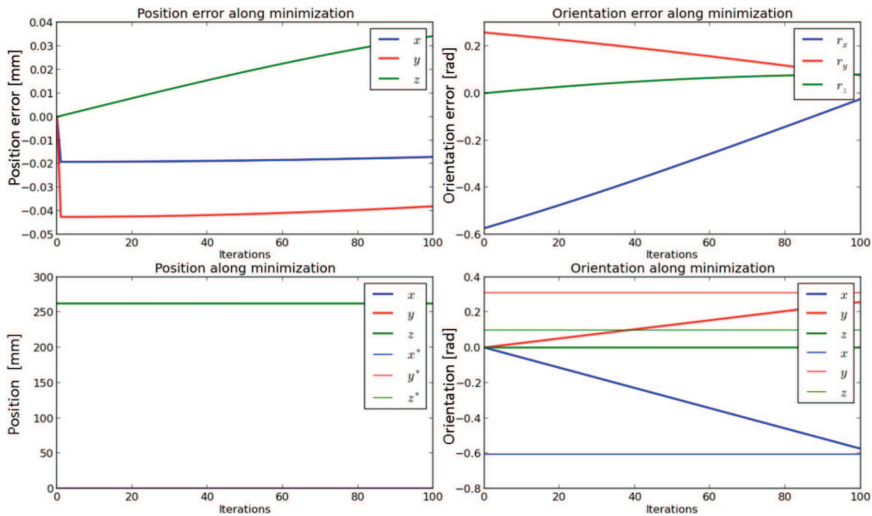
The next experiment requires mainly an orientation adjustment of the instrument tip, which can only be obtained by bending the modules. Figure 9.13 presents the initial and final configuration considered. The targeted motion involves a rotation of around 0.6 rad ( $34^\circ$ ) around the  $x$ -axis and 0.25 rad ( $14^\circ$ ) along the  $y$ -axis. In Figures 9.14 and 9.15, we can see that both module configuration encoding methods (tip position and chamber length respectively) allow performing the desired motion. In both cases, we can see some variation of the error in position, but the observed error values are quite small (maximum of 0.04 mm and 0.20 mm) and can be neglected.



**Figure 9.12** Performance of task shown in Figure 9.10 using the chamber length as model of a module configuration.

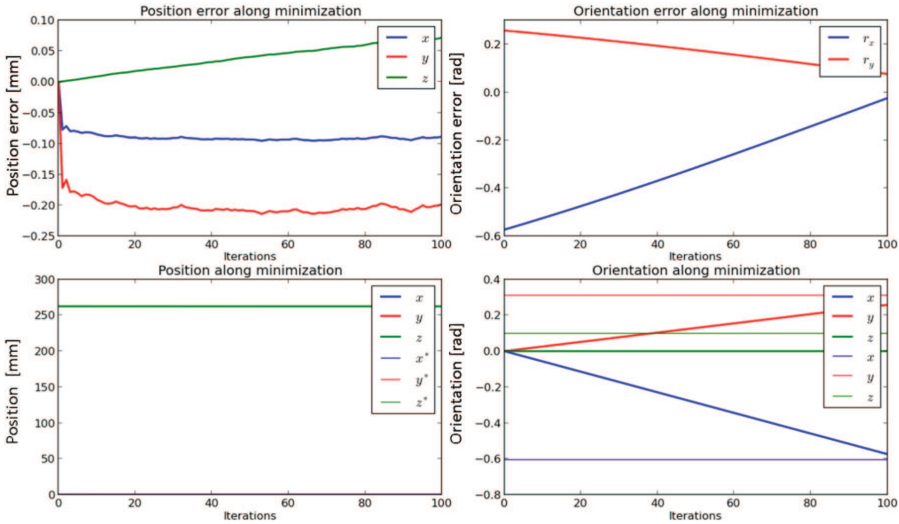


**Figure 9.13** Experiment 2: Motion experiment with three modules and a fixed base. (a) Initial position. (b) Target pose, requiring only adjustment of the tip orientation.



**Figure 9.14** Results of Experiment 2 using tip position as a feature.

Nevertheless, we can also observe that the complete convergence towards the desired orientation is not obtained. In the tip position case (Figure 9.14), an error of around 0.1 rad ( $5.7^\circ$ ) remains for the rotation around y-axis. A similar error along the x-axis is also observed for the case concerning the chamber length (Figure 9.15). On the one hand, we can consider that the



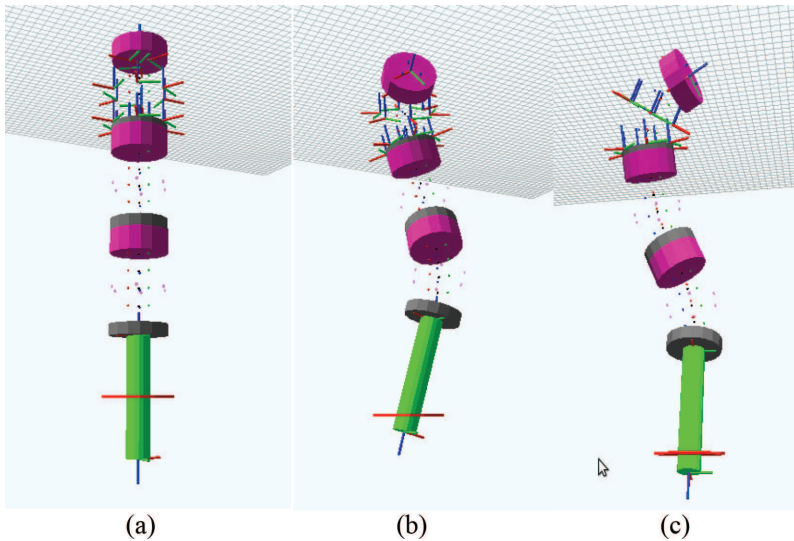
**Figure 9.15** Results of Experiment 2 using chamber length as a feature.

remaining error is quite small. On the other hand, such an experiment shows the limitations of the affine convergence model. Indeed, the error value is kept constant during the whole minimization process, and a perfect convergence can only be obtained if the Jacobian is well-conditioned. Perfect convergence may thus not be observed in the bounded number of iterations. Such an issue will be commented on again when discussing the next set of experiments.

Note, finally, that in the previous experiments, where the base of the structure is fixed, we do not control the error of orientation around the  $z$ -axis, since the modules cannot directly compensate such a motion. Therefore the error function is of dimension 5. All six dimensions are considered when the arm is mounted onto a moveable base, as presented in the experiments described in the next sections.

### 9.3.2 Inverse Kinematics Involving the Base under Single Point Insertion Constraint

In the following experiments, we consider that the flexible modules are fixed to a rigid component which itself is connected to a robotic arm, extending the motion capabilities of the whole system. In Figure 9.16, and in the following experiments, the green component represents the connecting component that would link the STIFF-FLOP arm to a standard robotic arm. In that case,



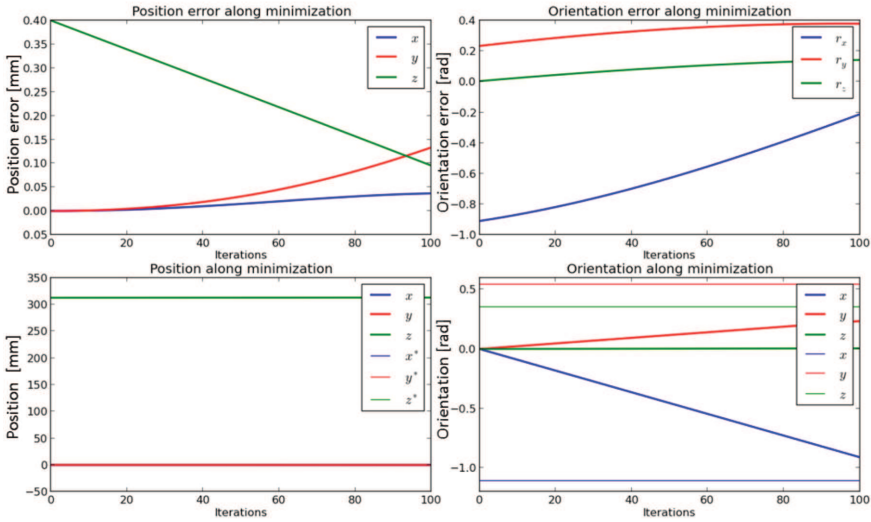
**Figure 9.16** Experiment 3: Inverse kinematics considering an additional four DOF by introducing a moveable base, through motion of the green rigid rod, under the insertion point constraint. (a) Initial position (b, c) two views of the desired configuration.

the end effector of the arm would be connected to the lower section of the green cylinder (as shown in the bottom of the figure). As previously described in section 9.1.2, the system is envisioned to enter the human body through an insertion point, and the produced motion should be so that the arm always respects the virtual pivot point related to the insertion frame. In Figure 9.16, the horizontal red line represents that critical location, being the image projection of the red disc presented previously in Figure 9.6.

In the following experiments, the inverse kinematics will deduce the appropriate configuration of the flexible modules as well as the appropriate pose of the green cylinder, expressed as an extended spherical coordinate of its tip section (connected to the modules) with respect to the insertion frame. We assume then that the robotic arm will produce the appropriate motion to move the whole structure according to the computed desired pose. Note that in the following experiments the flexible module configurations are modeled with their local tip position; i.e., the joint variable  $\mathbf{q}$  contains for each module its tip position  $Q_i$  expressed at its base.

The first experiment presented in Figure 9.16 is mainly a rotation motion. It is quite large as we can see in Figure 9.17: 0.9 rad ( $51^\circ$ ) around the  $x$ -axis and 0.2 rad ( $11^\circ$ ) around the  $y$ - and  $z$ -axes. Once more, the motion in position





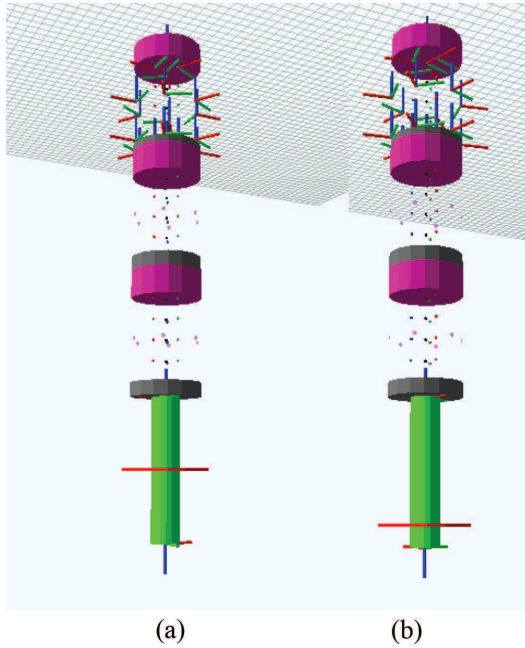
**Figure 9.17** Convergence details for Experiment 3 with the moving base and the flexible module configuration encoded with the tip position.

is negligible during the minimization process (maximum of 0.40 mm). The orientation globally converges towards the desired values; nevertheless, the desired orientation is not exactly reached within the specified number of iterations. We can indeed observe a remaining error of around 0.4 rad ( $23^\circ$ ) around the  $x$ -axis and 0.2 ( $11^\circ$ ) around the  $y$ - and  $z$ -axes.

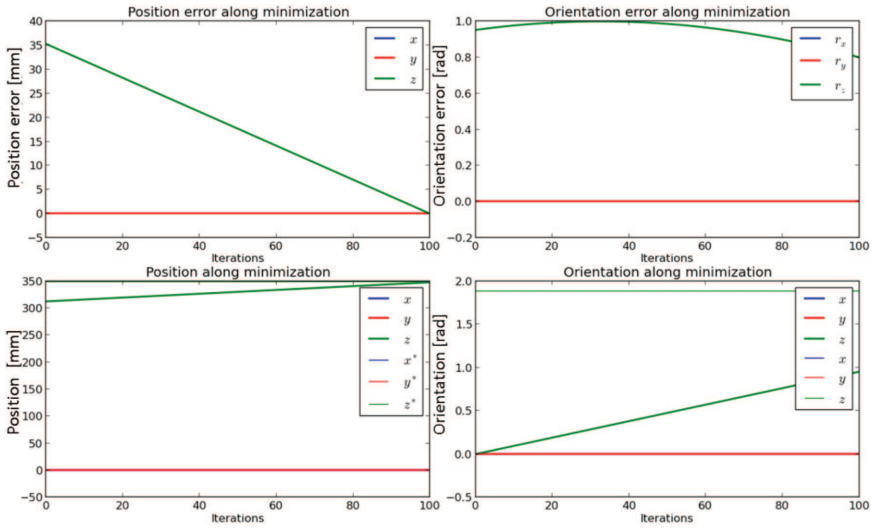
The non-completion of the task in the given timeframe is even more visible in the following experiment illustrated by Figures 9.18 and 9.19, in which the targeted displacement involves a large motion along the  $z$ -axis and a rotation around that same axis. In that case, the requested rotation is a bit more than 1.9 rad ( $109^\circ$ ).

If the error in position gets minimized on time, the error in rotation is not compensated on time. Once more the affine model shows its limit. Note that, when controlling both the base and module motions, different weights are applied to the different components, to give higher weight to the base. This indirectly affects the error minimized at each iteration and results in making the convergence impossible within the fixed number of iterations.

The most appropriate way to handle such an issue is to switch from the affine error model to an exponential model, in which the error observed is updated at each step and in which the number of iterations to convergence is not fixed. This is what has been introduced in the next experiments.



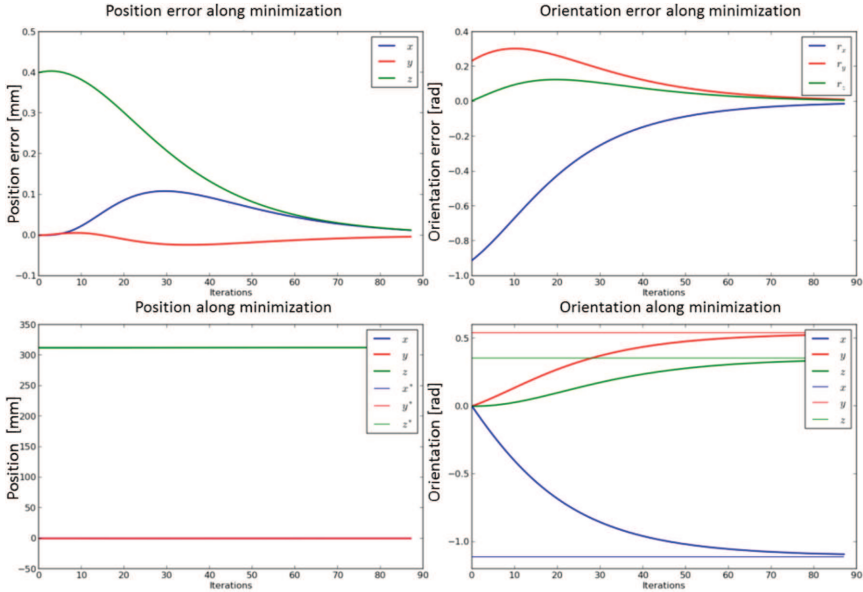
**Figure 9.18** Experiment 4: Motion task involving translation along the  $z$ -axis and rotation along the  $z$ -axis. (a) Initial position. (b) Target position.



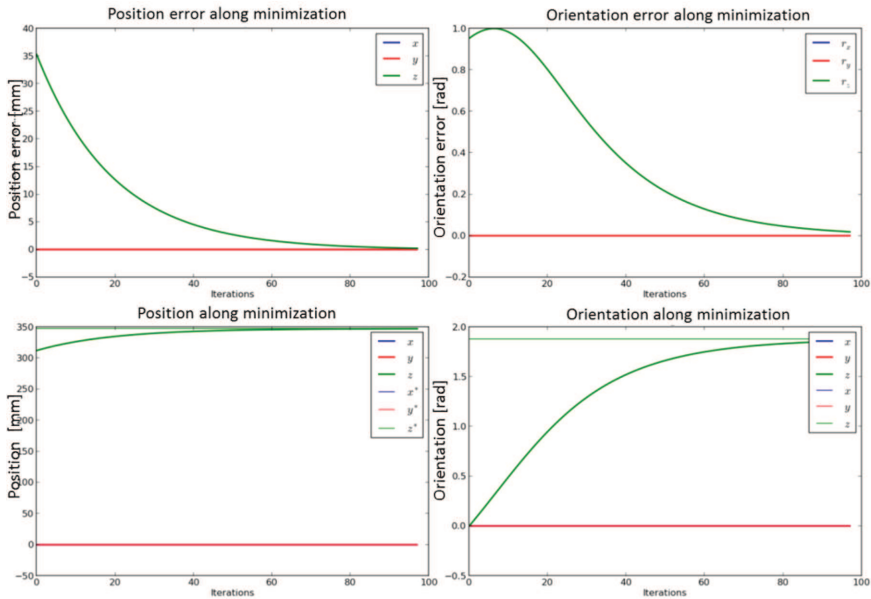
**Figure 9.19** Convergence details for Experiment 4.

As mentioned in section 9.2.1, different models of error evolution can be used as part of the minimization process. All previous experiments were done with an affine error decrease ( $\dot{e} = -\lambda$ ). Such an error model explains the evolution of the error along a straight line as observed in the previous convergence details presented. The following experiments are done considering an exponential decrease of the error ( $\dot{e} = -\lambda e$ ).

Figures 9.20 and 9.21, respectively, show the evolution of the error for the two experiments with the active base described earlier (Experiments 3 and 4) – however, this time we consider an exponential decrease of the error. In this case, the minimization process continues until the error reaches a given precision threshold or until a maximum number of iteration is reached. In both cases, we can observe the standard exponential evolution of the error. Furthermore, the system is now able to converge more precisely to the desired poses. In such a framework, particular care has to be taken when tuning the gain parameter. A high value permits quicker convergence, but may produce an oscillation around the desired pose. A small value may reduce or even avoid the potential oscillations, but may require a much larger number of iterations to converge.



**Figure 9.20** Convergence details for Experiment 3 (initial and desired poses presented in Figure 9.16), exponential evolution of the error.



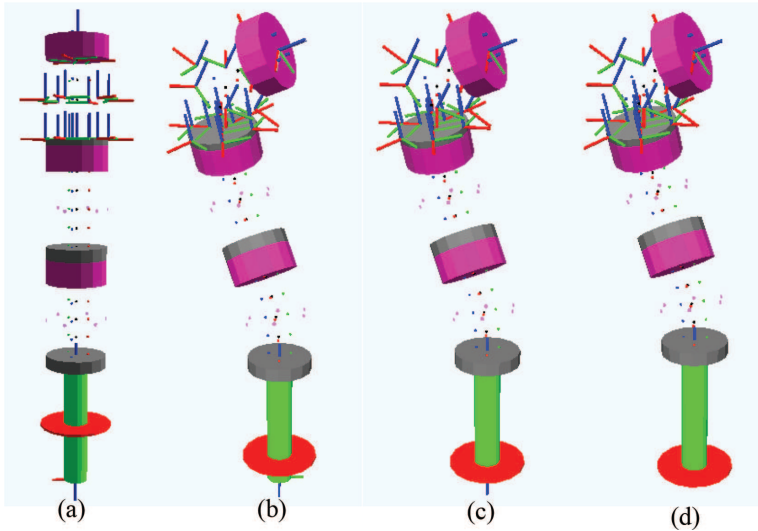
**Figure 9.21** Convergence details for Experiment 4 (Figure 9.18) with an exponential evolution of the error.

### 9.3.3 Illustration of the Secondary Tasks

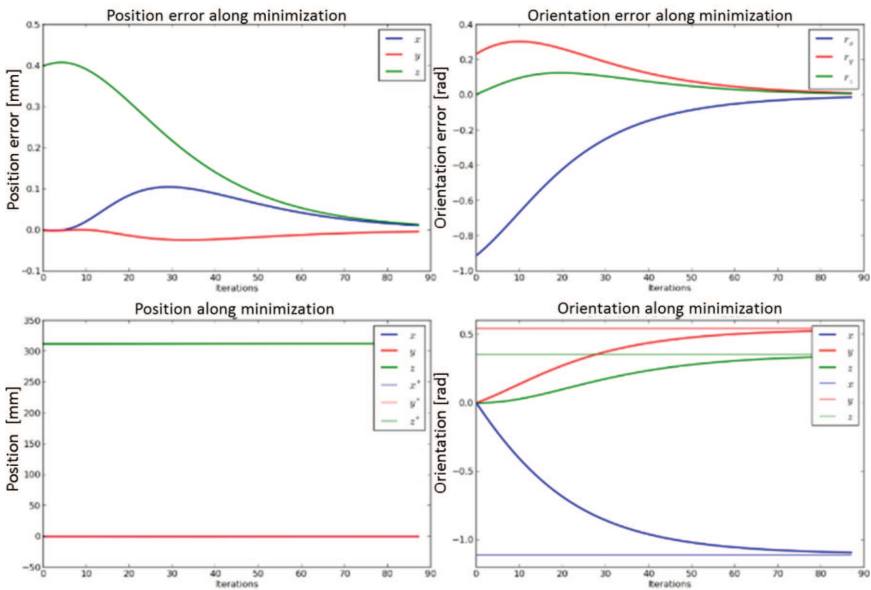
The following set of experiments illustrates the potential use of the system redundancy to control other constraints in the null space of the main task, which is the positioning one as described in section 9.2.5. The secondary tasks considered in the next experiments attempt to maintain the chambers at their mean length. Figure 9.22 presents the initial configuration and three final configurations after convergence towards the same target tip pose, but with different minimization settings for the secondary task. The three cases considered are the following ones:

- Case 1: No secondary task used,
- Case 2: Maximum elongation ( $\alpha$ ) set to 1,
- Case 3: Maximum elongation ( $\alpha$ ) set to 0.8.

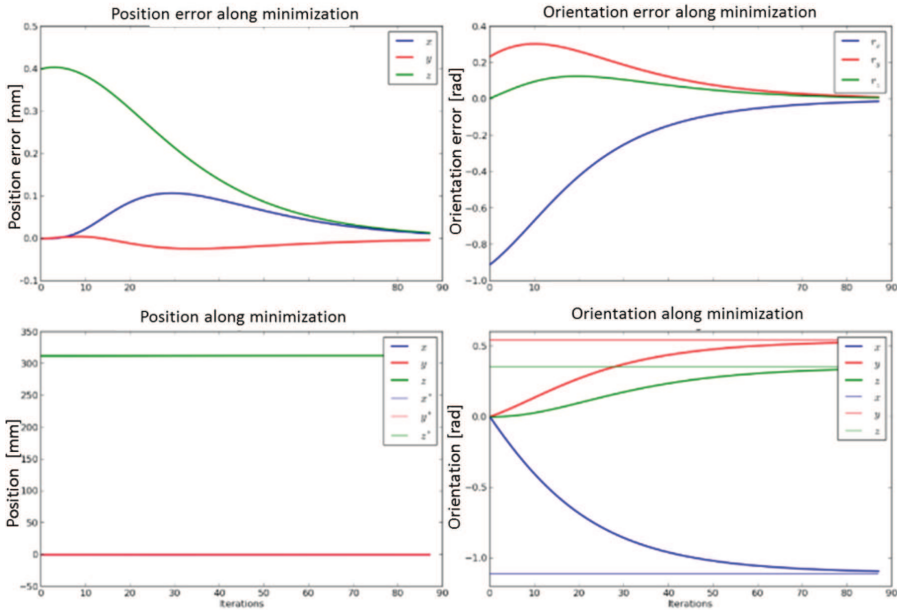
The first case (1) has already been presented, and the convergence details can be seen in Figure 9.20. For completion, the convergence data are presented in Figures 9.23 and 9.24 – it is noted that the overall behavior is very similar. As expected, the secondary task is applied in the space that does not affect the first one and is related to the control of the tip of the global structure.



**Figure 9.22** Experiment 4: Length control experimentation: (a) initial configuration, (b) obtained configuration after minimization without length control, (c) obtained configuration with first parameterization ( $\alpha = 1$ ), (d) obtained configuration with second parameterization ( $\alpha = 0.8$ ).



**Figure 9.23** Convergence details for Experiment 4 with length control, maximum elongation set to  $\alpha = 0.8$ .



**Figure 9.24** Convergence details for Experiment 4 with length control, maximum elongation set to  $\alpha=1.0$ .

Nevertheless, the final configurations of each module are slightly different, as can be seen from Table 9.1. The introduction of the secondary task to control the module length allows us to maintain the length of the modules to their mean dimensions as much as possible. Note that the proposed task function is designed with respect to the mean length of each module, and not directly with regard to the length of each chamber individually.

Since the modules extend their respective chambers less with regard to the overall elongation or bending of the robot, the base contributes more to the overall robot motion. This can be observed in Figure 9.22, by looking carefully at the location of the bottom extremity of the green rigid rod with respect to the red insertion disc. We can see that the base component is moving up more when the module length is being controlled. The reduced extension of the chamber length is being compensated by a larger motion of the base.

The next experiment illustrates the use of the secondary task for obstacle avoidance. The obstacle is avoided as soon as it is detected. The main

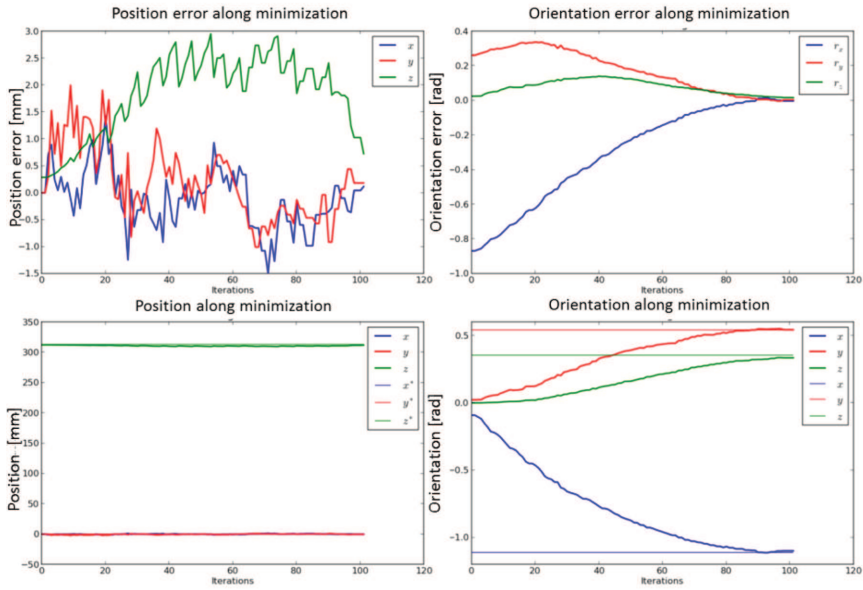
**Table 9.1** Comparison of the chamber lengths obtained using the secondary task (the length at rest is 0.04 m)

Experiment	Component	Module 1	Module 2	Module 3
Case 1 (no secondary task)	Chamber 1	0.067 m	0.057 m	0.059 m
	Chamber 2	0.065 m	0.065 m	0.055 m
	Chamber 3	0.059 m	0.076 m	0.076 m
	Mean extension	58%	58.33%	58%
Case 2 $\alpha = 1$	Chamber 1	0.06 m	0.049 m	0.052 m
	Chamber 2	0.057 m	0.058 m	0.047 m
	Chamber 3	0.051 m	0.061 m	0.069 m
	Mean extension	39.33%	39.66%	39.33%
Case 3 $\alpha = 0.8$	Chamber 1	0.064 m	0.053 m	0.056 m
	Chamber 2	0.061 m	0.062 m	0.052 m
	Chamber 3	0.055 m	0.065 m	0.073 m
	Mean extension	49.33%	49.33%	49.66%

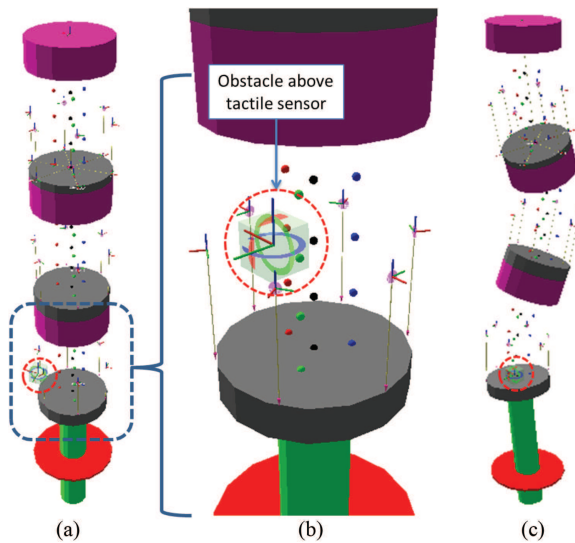
difference to the previous experiments is that the inverse kinematics is now computed in a closed-loop fashion: at each iteration, a motion request is sent to the module controllers, and the updated configuration is fed in to compute the next iteration of the minimization.

The experimental setting is similar to that of Figures 9.16 and 9.17. The convergence details obtained for this closed-loop mode, when no obstacle is detected, are presented in Figure 9.25. We can see that the trajectories are less smooth. This is mainly due to closed-loop implementation, since the dynamics of the controllers affect the minimization process (and the simulator may not be tuned sufficiently well to get a sufficiently reactive system). Nevertheless, the disturbances generated are quite small (magnitude of approximately 1 mm) and are mainly due to the fact that the emulation of the pressure variation as a function of time in the simulator may not be sufficiently well-tuned.

Figure 9.26 illustrates the settings of a similar experiment: an obstacle has been virtually placed above a tactile sensor and the system is required to move along the vertical  $z$ -axis. Figure 9.27 presents the behavior of the inverse kinematics in this case. The system moves freely along  $z$  until the tactile sensor detects the obstacle (iteration 10). The motion variation from iteration 10 to iteration 37 is related to the contribution of the obstacle avoidance task that pushes the robot modules away from the obstacle. Once avoided, the system is still able to converge towards the desired pose of the distal robot element, as we can see at the right snapshot of Figure 9.26.

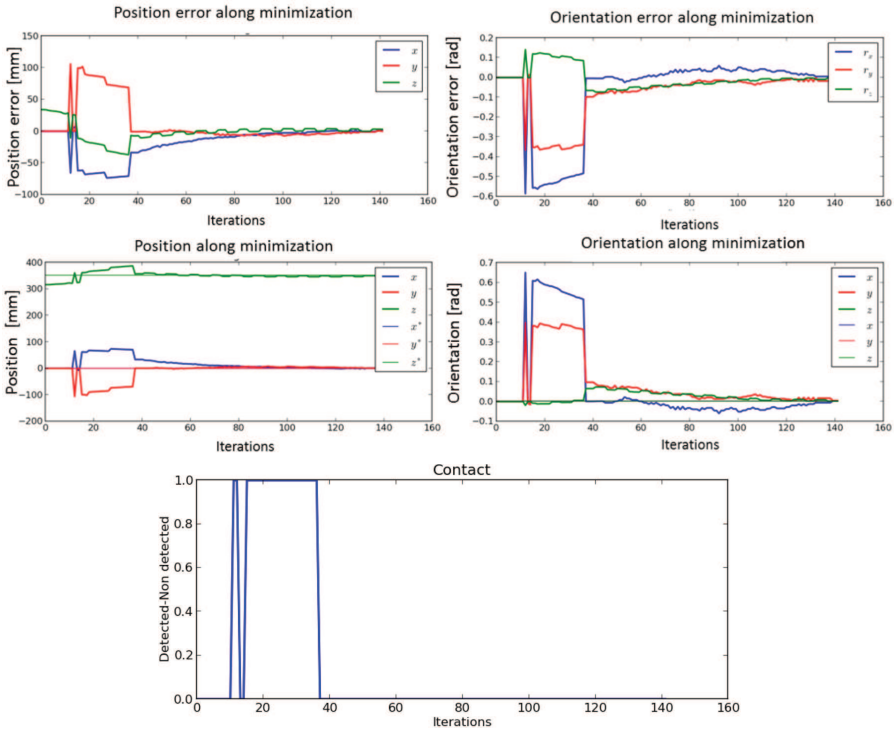


**Figure 9.25** Convergence details using a closed-loop mechanism. No obstacle avoidance activated.



**Figure 9.26** Target reaching with obstacle avoidance activated. (a) Initial configuration. An obstacle is placed just above a tactile sensor. (b) Zoom in on to the first module, the obstacle is represented through the cube. (c) Final configuration after convergence.





**Figure 9.27** Minimization with the obstacle avoidance constraint. The lower graphic presents the activation of the secondary task, when the contact is detected.

## 9.4 Conclusion

This section has presented how the flexible modules can be automatically controlled to reach a target tip pose provided by the surgeon operator through a regular haptic device. Contrary to the standard configuration space using the constant curvature model, we demonstrate that the generic inverse kinematic framework proposed, based on a numerical estimation of the robot Jacobian, is able to consider other configuration spaces, such as the module tip pose or directly the chamber length. Such a framework could even be used for taking into account flexible module models that do not assume the constant curvature preservation, which seems necessary for taking into account external forces acting on the modules.

We have also shown how the inclusion of an additional standard robot could be used to extend the work space, and how it can be easily added to the

inverse kinematics model. Assuming that such a robot provides position control of its end effector, we propose to embed it within the inverse kinematics framework using extended spherical coordinates that *per se* are suitable for representing the motion constraint due to the trocar location. We also mention that the traditional spherical model can be adjusted to be better shaped for surgical application, and without adding any complexity to the minimization process.

Finally, we illustrate with two specific tasks that the redundancy of the system can be used for applying, like with standard rigid-link robots, additional tasks to improve the behavior of the robot, while keeping the priority on the positioning task, that is the main request of the surgeon operator.

## References

- [1] Waldron, K., and Schmedeler, J. (2008). “Kinematics,” in *Springer Handbook of Robotics* (Berlin: Springer), 9–33.
- [2] Webster, R. J., and Jones, B. A. (2010). Design and kinematic modeling of constant curvature continuum robots: a review. *International Journal of Robotic Research*, 29, 1661–1683.
- [3] Calinon, S., Bruno, D., Malekzadeh, M. S., Nanayakkara, T., and Caldwell, D. G. (2014). Human-robot skills transfer interfaces for a flexible surgical robot. *Computer Methods and Programs in Biomedicine*, 116, 81–96.
- [4] Guthart, G. S., and Salisbury, K. J. (2000). “The intuitive telesurgery system: overview and application,” in *Proceedings of the IEEE International Conference on Robotics Automation*, Vol. 1, 618–621.
- [5] Zemiti, N., Morel, G., Ortmaier, T., and Bonnet, N. (2007). “Mechatronic design of a new robot for force control in minimally invasive surgery,” in *Proceedings of the IEEE/ASME Transactions on Mechatronics*, 12, 143–153.
- [6] Unger, S. W., Unger, H. M., and Bass, R. T. (1994). AESOP robotic arm. *Surgical Endoscopy*, 8:1131.
- [7] Ghodoussi, M., Butner, S. E., and Wang, Y. (2002). “Robotic surgery – The transatlantic case,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2, 1882–1888.
- [8] Aghakhani, N., Geravand, M., Shahriari, N., Vendittelli, M., and Oriolo, G. (2013). “Task control with remote center of motion constraint for minimally invasive robotic surgery,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, 5807–5812.

- [9] Nasserri, M. A., Gschirr, P., Eder, M., Nair, S., Kobuch, K., Maier, M., et al. (2014). “Virtual fixture control of a hybrid parallel-serial robot for assisting ophthalmic surgery: an experimental study,” in *Proceedings of the 5th IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechanics*, 732–738.
- [10] Ma, J., and Berkelman, P. (2006). “Control software design of a compact laparoscopic surgical robot system,” in *Proceedings of the 2006 IEEE/RSJ International Conference Intell Robot Systems*, 2345–2350.
- [11] Osa, T., Staub, C., and Knoll, A. (2010). “Framework of automatic robot surgery system using visual servoing,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1837–1842.
- [12] Staub, C., Panin, G., Knoll, A., and Bauernschmitt, R. (2010). Visual instrument guidance in minimally invasive robot surgery. *International Journal on Advances in Life Sciences*, 2, 103–114.
- [13] Muñoz, V. F., Gómez-de-Gabriel, J. M., García-Morales, I., Fernández-Lozano, J., and Morales, J. (2005). Pivoting motion control for a laparoscopic assistant robot and human clinical trials. *Advanced Robotics*, 19, 694–712.
- [14] Uecker, D. R., Lee, C., Wang, Y. F., and Wang, Y. (1995). Automated instrument tracking in robotically assisted laparoscopic surgery. *Journal of Image Guided Surgery*, 1, 308–325.
- [15] Chen, G. (2005). *Design, Modeling and Control of a Micro-robotic Tip for Colonoscopy*. Ph.D., thesis, Institut National des Sciences Appliquées, Lyon.
- [16] Fraś, J., Czarnowski, J., Maciaś, M., and Głowka, J. (2014). “Static modeling of multisection soft continuum manipulator for stiff-flop project,” in *Proceedings of the Advances in Intelligent Systems and Computing*, (Cham: Springer), 365–375.
- [17] Remazeilles, A., Prada, M., and Rasines, I. (2017). “Appropriate spherical coordinate model for trocar port constraint in robotic surgery,” in *Proceedings of the Iberian Robotics Conference*.

