

# A Parallel Manipulator for Prostate Cancer Biopsies: Kinematic Optimization, Design and Control Planning

Catarina Carvalheiras  
catarina.carvalheiras@tecnico.ulisboa.pt

Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal

June 2017

## Abstract

The development of a prototype of a parallel robot is presented in this work. This prototype is to be implemented with a trans-rectal ultra sound probe for prostate biopsies, which is to be co-manipulated by an urologist. Prostate cancer is the sixth most common cause of cancer death overall, representing 9% of all male deaths. The most common methods for diagnosis are blood tests for scanning of the prostate-specific antigen levels, and digital rectal examination. If these tests reveal some signs of cancer suspicion, an ultra sound, manually guided biopsy to the prostate is usually advised. In many cases, the doctor can miss the cancerous tissue which will lead to a false negative test. The objective of the developed device is to guide the doctor to the optimal points of perforation through trajectory control. This work focuses on the data-based kinematic optimization of an existing commercial haptic controller, the Novint Falcon. This device was disassembled, optimized and rebuilt. The objectives of optimization were the maximization of kinematic manipulability and largest number of trajectory points reached from data sets acquired during previously registered prostate biopsies. The geometric, kinematic and dynamic models are deduced. The design after optimization, manufacture and assembly procedures are detailed. In order to achieve controllable mechanics, the implementation of DC motors, rotation encoders with disks, signal filtering and processing, and serial communication using an Arduino board and Simulink software are described. A functional prototype was successfully built.

**Keywords:** Prostate Biopsy, Co-Manipulation, Parallel Robotics, Kinematic Modelling, Kinematic Optimization.

## 1. Introduction

The prostate is a gland of the male reproductive system. As an aging effect, the size and texture of the prostate gradually changes. It is common for men after 50 years of age to develop prostate problems and it is necessary to make tests in order to diagnose prostate cancer. These are: urine tests, prostate specific antigen (PSA) blood tests, digital rectal examination and trans-rectal biopsies guided by ultra-sound (TRUS). In his Master thesis dissertation, Tavares [1] reconstructed a 3D model of the prostates of 22 TRUS-guided biopsy patients who had previously been submitted to an magnetic resonance imaging (MRI). The data collected was analysed in order to determine what was the real position and relative displacement of the prostate during each one of the perforations.

It became clear that the perception of the Urologist to where he thought the sample acquisitions were realised was very different from the real position. In many cases the doctor can miss the cancer-

ous tissue which will lead to a false negative test. This represents a daily problem for urologists managing the disease, creates uncertainty and emotional stress for patients.

There is a noticeable advantage of cooperation between automated mechanisms and the knowledge of the human physicist. In this work, a step further is taken in human-robot interaction. The development of a parallel robot is presented, to be implemented with a TRUS probe, which is to be manipulated by an urologist. Because these robots are complex and usually expensive to build from scratch, a *Novint Falcon<sup>TM</sup>* haptic controller acquired from Novint Technologies, Inc. is used as a prototype of the robot to be used. However, this device is disassembled and kinematically optimized to reach a proper workspace for the prostate biopsy test. Then, links with the new dimensions are designed and built for the prototype. The main advantage is that the power source and actuators remain the same as the original haptic controller.

## 2. Data Acquisition

### 2.1. Registration Procedure

The objective was to record the pose of the TRUS probe during each one of the exams. To do this, a *Polaris Spectra*<sup>®</sup> optical tracking system by Northern Digital Inc. was used. A marker tool compatible with this system was attached to the probe, which allowed the *Polaris* to capture the location and orientation of the probe. This setup is shown in Figure 1.



Figure 1: Data acquisition setup. [1]

The orientation of the probe was returned in quaternions. Because this cloud point data is needed for the optimization of a kinematic structure, it was necessary to do some pre-processing. First, the original data came in the referential of the *Polaris*, which is observing the origin of the reference frame of the marker. The marker tool used has the reference frame shown in Figure 2.

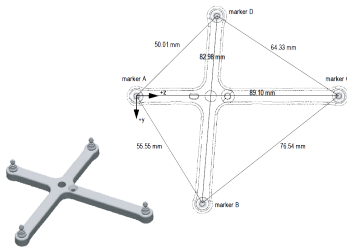


Figure 2: Marker tool

An average point and orientation were computed in order to project the data cloud. The quaternion parameters are defined as  $\mathbf{q} = [\eta, \epsilon_x, \epsilon_y, \epsilon_z]^T$ . Given two quaternions  $\mathbf{q}$  and  $\mathbf{q}_k$ , on the unit sphere  $S_3$  the Euclidean distance [2] is defined by:

$$\|\mathbf{q} - \mathbf{q}_k\| = \|\mathbf{I} - \mathbf{q}_k \mathbf{q}^{-1}\| = 2 \sin \frac{\eta_k}{4} \quad (1)$$

The square of Euclidean distance is related to the

scalar product of the quaternions such as:

$$\|\mathbf{q} - \mathbf{q}_k\|^2 = d_k^2 = 2[1 - (\eta \eta_k + \mathbf{q} \cdot \mathbf{q}_k)] \quad (2)$$

For a given set of  $N$  rotations, the average orientation is obtained by minimizing:

$$\min \frac{1}{N} \sum_{k=1}^N d_k^2 = \sum_{k=1}^N 2[1 - (\eta \eta_k + \mathbf{q} \cdot \mathbf{q}_k)] \quad (3)$$

subject to  $\|\mathbf{q}\|^2 = \eta \eta + \mathbf{q} \cdot \mathbf{q} = 1$

Which, after some Lagrange multipliers calculations follows that the mean orientation is given by the arithmetic average of the quaternions, normalized by the norm of the sum as:

$$\bar{\mathbf{q}} = \frac{(\mathbf{q}_1 + \mathbf{q}_2 + \dots + \mathbf{q}_N)}{\|\mathbf{q}_1 + \mathbf{q}_2 + \dots + \mathbf{q}_N\|} \quad (4)$$

For the cloud of points of all patients an average frame was obtained, and can be used as the reference frame of origin for the robot to be designed.

### 2.2. Workspace Characterization

The workspace needed for the manipulation of the probe during the biopsies can be characterized from the data sets. In Figure 3, it is plotted the points cloud of one patient and the average coordinate frame obtained.

The points registered for the probe are displayed such that the blue vectors indicate the longitudinal axis of the probe, and the green ones represent a perpendicular to the front side of the probe. The color scheme red, green, blue matches the x y z axis of the average frame, respectively. It is visible that the probe's displacement lies in an arched trajectory around the gluteus of the patient.

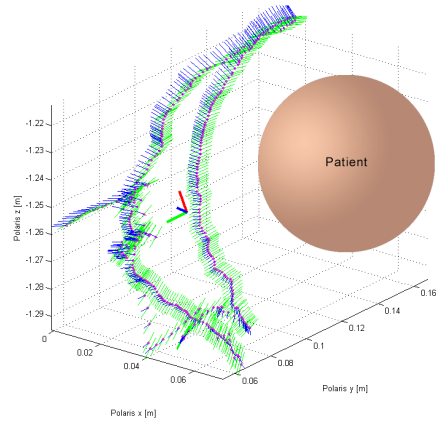


Figure 3: Points cloud of one patient.

### 3. Robot Modelling

A *Novint Falcon*<sup>TM</sup> haptik controller acquired from Novint Technologies, Inc. as the one shown in Figure 4 is to be kinematically optimized.

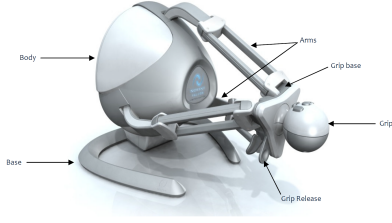


Figure 4: *Novint Falcon*<sup>TM</sup> haptik controller.

#### 3.1. Geometric Model

The geometric model will be deduced in the inverse form. This demonstration was based on the approaches of modelling done by Stamper [3] for a simpler 3 DOF parallel manipulator and Shah et al. [4]. A diagram of the geometry of one of the legs of the Falcon is shown in Figure 5.

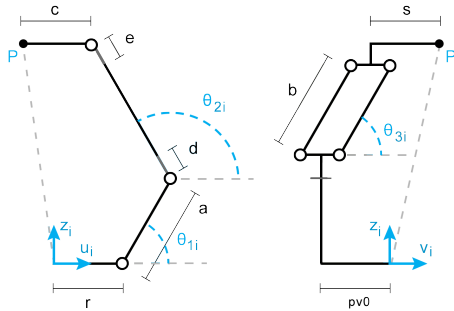


Figure 5: Representation of *i*th falcon leg.

The position of point  $\mathbf{P}_{uvw} = [p_{ui} \ p_{vi} \ p_{wi}]^T$ , which is the base of the end effector, is described in the  $UVW$  coordinate frame of the *i*th leg:

$$p_{ui} = a \cos \theta_{1i} - c + \cos \theta_{2i} (2d + b \sin \theta_{3i}) \quad (5)$$

$$p_{vi} = b \cos \theta_{3i} + s \quad (6)$$

$$p_{wi} = a \sin \theta_{1i} + \sin \theta_{2i} (2d + b \sin \theta_{3i}) \quad (7)$$

It is also needed to consider the position and orientation of each one of the leg's reference frames  $UVW$ , in the robot frame  $XYZ$  as represented in Figure 6. A transformation of the point  $\mathbf{P}_{xyz} = [p_x \ p_y \ p_z]^T$  on the global frame can be written as:

$$\begin{bmatrix} p_{ui} \\ p_{vi} \\ p_{wi} \end{bmatrix} = \begin{bmatrix} \cos(\phi_i) & \sin(\phi_i) & 0 \\ -\sin(\phi_i) & \cos(\phi_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} + \begin{bmatrix} -r \\ p_{v0} \\ 0 \end{bmatrix} \quad (8)$$

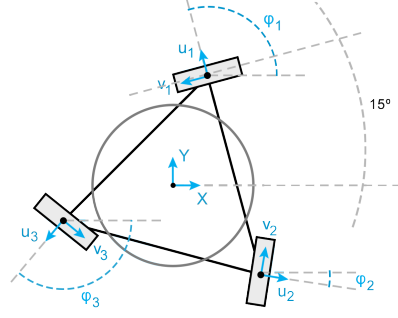


Figure 6: Base geometry of the falcon.

#### 3.1.1 Inverse Geometric Model

The solution for the third joint angle is:

$$\theta_{3i} = \arccos\left(\frac{p_{vi} - s}{b}\right) \quad (9)$$

Defining an auxiliary half-angle tangent, the following relationships can be written.

$$t_{1i} = \tan\left(\frac{\theta_{1i}}{2}\right) \leftrightarrow \begin{cases} \sin(\theta_{1i}) = \frac{2t_{1i}}{1+t_{1i}^2} \\ \cos(\theta_{1i}) = \frac{1-t_{1i}^2}{1+t_{1i}^2} \end{cases} \quad (10)$$

$$l_{2i}t_{1i}^2 + l_{1i}t_{1i} + l_{0i} = 0, \quad \text{where} \quad (11)$$

$$\begin{cases} l_{0i} = p_{wi}^2 + p_{ui}^2 + 2cp_{ui} - 4d^2 \\ \quad - b^2(\sin(\theta_{3i}))^2 - 4bd \sin(\theta_{3i}) - 2ap_{ui} + (a-c)^2 \\ l_{1i} = -4ap_{wi} \\ l_{2i} = p_{wi}^2 + p_{vi}^2 + 2cp_{vi} - 4d^2 \\ \quad - b^2(\sin(\theta_{3i}))^2 - 4bd \sin(\theta_{3i}) + 2ap_{vi} + (a+c)^2 \end{cases}$$

A solution for  $\theta_{1i}$  can be found from 11 with data from the end effector position:

$$\theta_{1i} = 2 \arctan(t_{1i}), \quad \text{where} \quad t_{1i} = \frac{-l_{1i} - \sqrt{(l_{1i}^2 - 4l_{2i}l_{0i})}}{2l_{2i}} \quad (12)$$

Knowing the value of  $\theta_{1i}$  and combining 5 and 7, one can write  $\theta_{2i}$  such that:

$$\theta_{2i} = \arctan\left(\frac{p_{wi} - a \sin(\theta_{1i})}{p_{ui} - a \cos(\theta_{1i}) + c}\right) \quad (13)$$

#### 3.2. Kinematic Model

To compute the kinematic model, a geometric jacobian  $\mathbf{J}$  is to be defined. Equations 5 to 7 will be differentiated with respect to time, which represent the velocity of point  $\dot{\mathbf{P}}_{uvw} = [\dot{p}_{ui} \ \dot{p}_{vi} \ \dot{p}_{wi}]^T$ , in each leg's coordinate system.

### 3.2.2 Inverse Kinematic Model

$$\begin{aligned} \dot{p}_{ui} = & - \left( a \sin \theta_{1i} \right) \dot{\theta}_{1i} - \left( \sin \theta_{2i} (2d + b \sin \theta_{3i}) \right) \dot{\theta}_{2i} + \\ & + \left( b \cos \theta_{2i} \cos \theta_{3i} \right) \dot{\theta}_{3i} \end{aligned} \quad (14)$$

$$\dot{p}_{vi} = - \left( b \sin \theta_{3i} \right) \dot{\theta}_{3i} \quad (15)$$

$$\begin{aligned} \dot{p}_{wi} = & \left( a \cos \theta_{1i} \right) \dot{\theta}_{1i} + \left( \cos \theta_{2i} (2d + b \sin \theta_{3i}) \right) \dot{\theta}_{2i} + \\ & + \left( b \sin \theta_{2i} \cos \theta_{3i} \right) \dot{\theta}_{3i} \end{aligned} \quad (16)$$

Solving equation 15 for  $\dot{\theta}_{3i}$ , one obtains:

$$\dot{\theta}_{3i} = - \frac{\dot{p}_{vi}}{b \sin(\theta_{3i})} \quad (17)$$

Substituting Equation 17 in Equations 14 and 16, it is possible to remove  $\dot{\theta}_{3i}$ . To eliminate  $\dot{\theta}_{2i}$ , the results are combined in Equation 18, referred to as  $f_i$ .

$$\begin{aligned} \dot{p}_{ui} \left( \sin \theta_{3i} \cos \theta_{2i} \right) + \dot{p}_{vi} \left( \cos \theta_{3i} \right) + \\ + \dot{p}_{wi} \left( \sin \theta_{3i} \sin \theta_{2i} \right) + \dot{\theta}_{1i} \left( a \sin(\theta_{1i} - \theta_{2i}) \sin \theta_{3i} \right) \end{aligned} \quad (18)$$

Differentiating 8 with respect to time, one obtains 19 as a function of the velocity of point  $\dot{\mathbf{P}}_{xyz} = [\dot{p}_x \ \dot{p}_y \ \dot{p}_z]^T$ .

$$\begin{aligned} \begin{bmatrix} \dot{p}_{ui} \\ \dot{p}_{vi} \\ \dot{p}_{wi} \end{bmatrix} = \begin{bmatrix} \cos(\phi_i) & \sin(\phi_i) & 0 \\ -\sin(\phi_i) & \cos(\phi_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{p}_z \end{bmatrix} \\ \Leftrightarrow \dot{\mathbf{P}}_{uvw} = \mathbf{T}_{xyz}^{uvw} \dot{\mathbf{P}}_{xyz} \end{aligned} \quad (19)$$

#### 3.2.1 Geometric Jacobian

Computing the partial derivatives of Equation  $f_i$  (18) with respect to  $\dot{\mathbf{P}}_{uvw}$  and  $\dot{\theta}_1$ , one obtains  $\mathbf{A}_{r_{uvw}}$  and  $\mathbf{B}$ . The relation between actuated joint velocities  $\dot{\theta}_1 = [\dot{\theta}_{11} \ \dot{\theta}_{12} \ \dot{\theta}_{13}]^T$  and end effector velocity  $\dot{\mathbf{P}}_{xyz}$  is:

$$\mathbf{A}_{r_{uvw}} \mathbf{T}_{xyz}^{uvw} \dot{\mathbf{P}}_{xyz} = \mathbf{B} \dot{\theta}_1 \quad (20)$$

Thus, the jacobian,  $\mathbf{A}_r$  and  $\mathbf{B}$  can be defined by:

$$\mathbf{J} = \left( \mathbf{A}_{r_{uvw}} \mathbf{T}_{xyz}^{uvw} \right)^{-1} \mathbf{B} \quad \Leftrightarrow \quad \mathbf{J} = \left( \mathbf{A}_r \right)^{-1} \mathbf{B} \quad (21)$$

$$\mathbf{A}_r = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} B_1 & 0 & 0 \\ 0 & B_2 & 0 \\ 0 & 0 & B_3 \end{bmatrix} \quad (22)$$

$$A_{i1} = \cos(\theta_{2i}) \sin(\theta_{3i}) \cos(\phi_i) - \cos(\theta_{3i}) \sin(\phi_i) \quad (24)$$

$$A_{i2} = \cos(\theta_{2i}) \sin(\theta_{3i}) \sin(\phi_i) + \cos(\theta_{3i}) \cos(\phi_i) \quad (25)$$

$$A_{i3} = \sin(\theta_{2i}) \sin(\theta_{3i}) \quad (26)$$

$$B_i = a \sin(\theta_{2i} - \theta_{1i}) \sin(\theta_{3i}) \quad (27)$$

If the position and acceleration of the end effector are known, the solution for the joint angle accelerations  $\ddot{\theta}_1 = [\ddot{\theta}_{11} \ \ddot{\theta}_{12} \ \ddot{\theta}_{13}]^T$  can be found as a reduced Inverse Kinematic Model:

$$\ddot{\theta}_1 = \mathbf{J}_{inv} \ddot{\mathbf{P}}_{xyz} \quad (28)$$

$$\mathbf{J}_{inv} = \left( \mathbf{B} \right)^{-1} \mathbf{A}_r \quad (29)$$

### 3.3. Kinematic Model of Second Order

Reversing 28 and substituting the jacobian:

$$\mathbf{A}_r \ddot{\mathbf{P}}_{xyz} = \mathbf{B} \ddot{\theta}_1 \quad \Leftrightarrow \quad \ddot{\mathbf{P}}_{xyz} = \mathbf{J} \ddot{\theta}_1 \quad (30)$$

It is necessary to differentiate this forward kinematic equation with respect to time, such that:

$$\dot{\ddot{\mathbf{P}}}_{xyz} = \mathbf{J} \ddot{\theta}_1 + \dot{\mathbf{J}} \dot{\theta}_1 \quad (31)$$

The computation of  $\dot{\mathbf{J}}$  is detailed in Equation 32.

$$\begin{aligned} \dot{\mathbf{J}} = \frac{\mathbf{B}}{\det^2(\mathbf{A}_r)} \left( \frac{d}{dt} \text{adj}(\mathbf{A}_r) \det(\mathbf{A}_r) - \right. \\ \left. \text{adj}(\mathbf{A}_r) \frac{d}{dt} \det(\mathbf{A}_r) \right) + \frac{\dot{\mathbf{B}}}{\det(\mathbf{A}_r)} \text{adj}(\mathbf{A}_r) \end{aligned} \quad (32)$$

Where  $\text{adj}(A_r)$  is the adjoint matrix of  $A_r$  and  $\det(A_r)$  its determinant.

### 3.4. Dynamic Model

The free body diagram for the  $i$ th leg can be seen in Figure 7.  $W$  represents the weight of the end effector platform while  $h_p$  is the torque in the first joint caused by the forces applied at the end effector P. Two assumptions must be defined:

1. The base of the robot is on the ground and gravity acts in the negative global Z direction.
2. The mass  $m_b$  in link b, is evenly divided and concentrated at joints  $B_i$  and  $E_i$ .

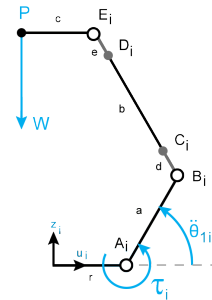


Figure 7: Free body diagram of  $i$ th leg.

One can write two equations for the sum of moments at the first joint:

$$\sum M_{A_i} = I_A \ddot{\theta}_{1i} + c_d \dot{\theta}_{1i} + h_p \quad (33)$$

$$\sum M_{A_i} = \tau_i - ag \cos(\theta_{1i}) \left( \frac{1}{2} m_a + m_b \right) - \mathbf{J}^T m_p \mathbf{g} \quad (34)$$

$$I_A = I_m + a^2\left(\frac{1}{3}m_a + m_b\right) \quad (35)$$

$$h_p = \mathbf{J}^T m_p \ddot{\mathbf{p}} \quad (36)$$

$$m_p = 3m_b + m_c \quad (37)$$

$$\mathbf{g} = [0 \quad 0 \quad g]^T \quad (38)$$

$g \Rightarrow$  Gravitational acceleration;

$I_A \Rightarrow$  Mass moment of inertia of link a;

$I_m \Rightarrow$  Mass moment of inertia of the motor rotor;

$m_a \Rightarrow$  Mass of link a;

$m_b \Rightarrow$  Mass of each rod of link b;

$m_c \Rightarrow$  Mass of the platform and payload;

$c_d \Rightarrow$  Viscous damping coefficient of the actuator;

$\tau_i \Rightarrow$  Torque applied by the actuator

$\mathbf{J} \Rightarrow$  Jacobian matrix from Equation 21;

$\ddot{\mathbf{p}} \Rightarrow$  Acceleration of the end effector.

Combining Equation 33 and 34 for each leg, Equations 35 to 38 and 31 are substituted in the final matricial expression for all actuated joints.

$$\tau = \mathbf{M}(\theta)\ddot{\theta} + \mathbf{C}(\theta, \dot{\theta})\dot{\theta} + \mathbf{G}(\theta) \quad (39)$$

Where  $\tau = [\tau_1 \quad \tau_2 \quad \tau_3]^T$ ,  $\mathbf{I}$  is the identity and,

$$\mathbf{M}(\theta) = \left(I_m + a^2\left(\frac{m_a}{3} + m_b\right)\right) \mathbf{I} + \mathbf{J}^T (3m_b + m_c) \mathbf{J} \quad (40)$$

$$\mathbf{C}(\theta, \dot{\theta}) = c_d \mathbf{I} + \mathbf{J}^T (3m_b + m_c) \dot{\mathbf{J}} \quad (41)$$

$$\mathbf{G}(\theta) = ag\left(\frac{1}{2}m_a + m_b\right) \begin{bmatrix} \cos(\theta_{11}) \\ \cos(\theta_{12}) \\ \cos(\theta_{13}) \end{bmatrix} + \mathbf{J}^T (3m_b + m_c) \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \quad (42)$$

#### 4. Kinematic Optimization

The criteria of optimization will be the maximum manipulability and number of points reached from the cloud point data.

##### 4.1. Manipulability

The manipulability is defined by the capacity of changing the position and orientation of the end effector given a joint configuration. This capability is described by the velocity manipulability ellipsoid [5] represented in Figure 8. Considering the euclidean norm of joint velocities

$$\|\dot{\mathbf{q}}\| = \sqrt{\dot{q}_1^2 + \dot{q}_2^2 + \dots + \dot{q}_n^2} \quad \text{such that} \quad \|\dot{\mathbf{q}}\| \leq 1, \quad (43)$$

the set of robot velocities  $\dot{\mathbf{x}}$  which is realized by equation 43, is the manipulability ellipsoid [6]. The principal axes' directions of the manipulability ellipsoid will be the eigenvalues  $\lambda_i$  of  $\mathbf{J}\mathbf{J}^T$ , while their dimensions will be the singular values  $\sigma_i = \sqrt{\lambda_i}$ .

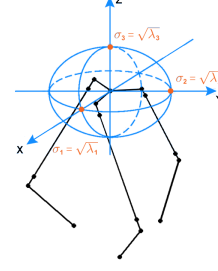


Figure 8: Manipulability ellipsoid representation.

A global representative measure of manipulability can be obtained by considering the volume of the ellipsoid. This volume is proportional to  $w(q)$ , detailed in equation 44.

$$w(q) = \sqrt{\det(\mathbf{J} \mathbf{J}^T)} \quad (44)$$

##### 4.2. Parameters

The geometric parameters of the Novint Falcon are specified in Table 1 [7] as represented in Figure 5.

Parameter	Value
a [mm]	60
b[mm]	103
d=e [mm]	12
r [mm]	37
pv0 [mm]	23
c [mm]	16
s [mm]	-25
$\phi_1$ [rad]	$\frac{7\pi}{12}$
$\phi_2$ [rad]	$\frac{-\pi}{12}$
$\phi_3$ [rad]	$\frac{-9\pi}{12}$

Table 1: Geometric parameters. [7]

It was noticeable that there was few relevance of changing parameters  $c$ ,  $s$ ,  $\phi_1$ ,  $\phi_2$  and  $\phi_3$ . Iso, the orientation of the coordinate frames of each leg was kept such as to maintain the overall symmetry of the structure. The parameters to optimize are:

$$\text{Parameters} = [a, b, d, r, pv0, z] \quad (45)$$

The base reference frame of the robot will be assigned to the average frame computed from the points cloud. However, the first reachable points are a few centimetres apart from the base reference frame in the positive direction of  $Oz$  axis. This explains the need for parameter  $z$  to move the points cloud as illustrated in Figure 9. The major axes of

displacement in the robot are the  $Ox$  and  $Oy$  and, as such, they were aligned to match the biggest displacements needed in the points cloud in  $x$  and  $y$ .

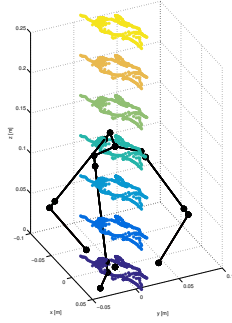


Figure 9: Increase in optimization parameter  $z$ .

### 4.3. Algorithm

The goal is to maximize the manipulability in the points obtained from the data acquisition. For this, Equation 43 is the cost function to maximize, applied to the inverse kinematic model of each one of the cloud points. The method chosen to realize this optimization provided a good control of the physical meaning of the optimization parameters. This approach was a discrete iterative search with a double refinement in the mesh values to check for convergence. The cost function is computed for all the biopsy points in one parameters' mesh value. Only after is the true cost function derived by the average of the manipulability measure for all points in this mesh value, that are within the workspace of the robot. This procedure is used to avoid local maxima in finding the maximum manipulability for all mesh values. In reality, a maximum average manipulability is optimized, and the number of points reached is accounted for. The algorithm implemented is described in Algorithm 1.

### 4.4. Results

The first run of the algorithm was an iteration where the set of values that each parameter could take are presented in Table 2. The results obtained for this first run, where a rough mesh was used, are represented in Figure 10. This picture represents the obtained set of parameters which maximize the average manipulability measure, for each value taken by the parameter  $a$ . Analysing the results, one can see that there were two mesh points where the average manipulability reached maximums. The measure is high for both sets, but while the first set reached only 10850 biopsy points out of 10954, the second set reached 10938, representing 99.85% of all the points cloud. Another iteration must be done with a more refined range of parameters in order to prove convergence of the algorithm.

The range of values that each parameter could take are again redefined and presented in Table 3. The results of the refinement are plotted in the graph of Figure 11.

#### Algorithm 1: Discrete Iterative Search

```

Input : Biopsy points cloud
Output: Parameters for the greatest Average
          Manipulability measure
1 For each  $a = \min a : \text{step} : \max a$  , do
2 For each  $b = \min b : \text{step} : \max b$  , do
3 For each  $d = \min d : \text{step} : \max d$  , do
4 For each  $r = \min r : \text{step} : \max r$  , do
5 For each  $pv0 = \min pv0 : \text{step} : \max pv0$  , do
6 For each  $z = \min z : \text{step} : \max z$  , do
7 Assign  $[a, b, d, r, pv0, z]$ ;
8 For each  $p = 1st p : \text{last } p$  , do
9  $p_z = p_z + z$ ;
10 IGM  $\theta = L^{-1}(\mathbf{p})$ ;
11 If Point is within workspace , then
12 Increment counter of points reached by
   these parameters;
13 Manipulability measure  $w(q)$  of this point
   for these parameters;
14 EndIf
15 EndFor
16 Compute Average Manipulability measure of
   the set of points;
17 If Average Manipulability measure is greater
   than previous , then
18 Save greatest Average Manipulability
   measure;
19 Save mesh point parameters;
20 EndIf
21 EndFor
22 EndFor
23 EndFor
24 EndFor
25 EndFor
26 EndFor

```

Param	Min	Max	Step
a	60	150	10
b	100	300	20
d=e	10	30	10
r	50	150	10
$\ pv0\ $	20	100	10
z	100	600	100

Table 2: Values for the first optimization.

After the analysis of the refinement results around the first original optimal point, the convergence of the search is proved. The maximum point of manipulability obtained was the same as in the first iteration, corresponding to  $a = 140$ ,  $b = 140$ ,  $d = 30$ ,  $r = 90$ ,  $pv0 = 50$ ,  $z = 10$ , in millimeters, with equal values of average manipulability measure and points reached. One can argue that a deeper refinement would be needed. However, because these parameters are to be applied in the design of the new structure, the tolerance of around 5mm in each parameter can be considered

acceptable. This because the methods of manufacture adopted to build the robot are not so precise. The relative proportions of the original robot and the optimized one can be compared in Figure 12(a), and the relative size of the workspace of both structures can be seen in Figure 12(b).

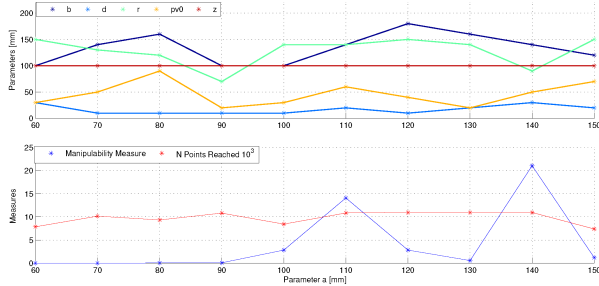


Figure 10: Results obtained for the first run. Top plot: Optimization parameters displayed in terms of the evolution of  $a$ , in the x axis. Bottom plot: Maximum average manipulability measure and points reached obtained for each set.

Param	Min	Max	Step
a	135	145	5
b	130	150	10
d=e	25	40	5
r	85	95	5
$\ pv0\ $	45	55	5
z	100	200	100

Table 3: Values for the refinement optimization.

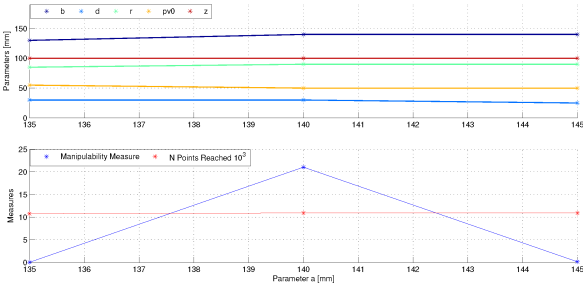


Figure 11: Results obtained for the 2nd iteration.

In green, the original workspace of the Falcon presents a range in the x and y axes of  $[-0.1; 0.1]$ m in a total of 20cm and in the z axis of  $[0; 0.15]$ m in a total of 15cm. In blue, the optimized workspace shows a wider range in all the axes: from  $[-0.15; 0.15]$ m in x and y, in a total of 30cm, and in the z axis of  $[0; 0.25]$ m, in a total of 25cm. It is visible that the workspace was stretched more in the positive z direction and fewer in x and y. This is coherent with the data since, during a biopsy, it is more necessary to move the probe in order to rotate the ultra-sound sensor in the sagittal plane of the patient, than to displace the sensor.

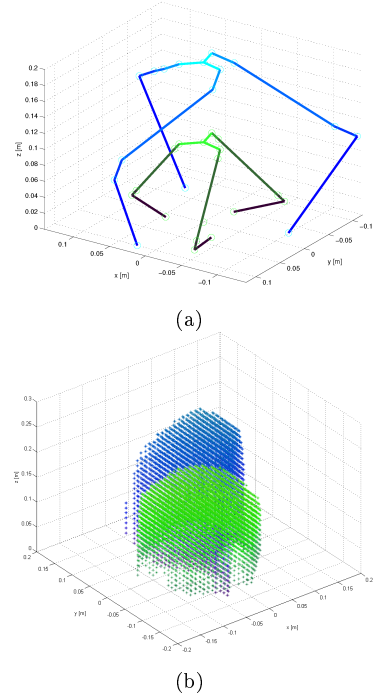


Figure 12: (a) Optimized structure in blue and the original one in green. (b) Workspace comparison.

## 5. Implementation

### 5.1. Design

After the kinematic optimization, the robot links need to be designed according to the obtained measures. The arm consists of: a circular link (1); an adapter from the first joint to the parallelogram structure (2); the two parallelogram bars (3); the adapter from the parallelogram to the end effector (4) and the end effector itself, as represented in Figure 13(a). To add the rotation of the TRUS probe to the 3 translational DOFs of the robot, it was necessary to create an interface for the end effector. This consists of a spheric passive joint, to be actuated by the urologist according to the necessary orientation for the probe, inside the patient. This adds 2 rotational DOFs to the robot, now with a total of 5DOFs. In order to follow the parameters obtained by the output of the optimization, there is the need to assemble the robot with some geometrical precision. The result was the use of inexpensive laser cutting, which enabled the fabrication of a support box for the robot. This encasing not only contains the desired holes to fixate the arms with respect to the geometric constraints, but also the motors and sensors.

### 5.2. Actuators

From data obtained from online reports [8], it was possible to know that the *Novint Falcon* was powered by a 30V, 1A transformer. From this information, one can extrapolate that the maximum volt-

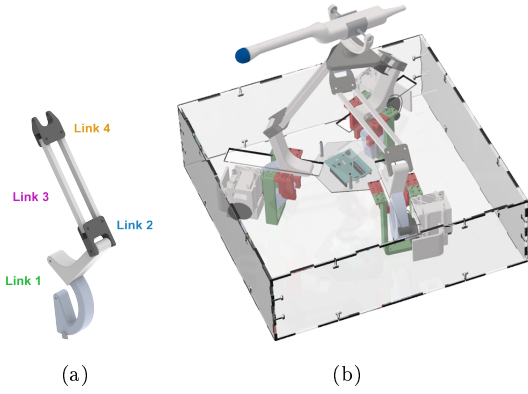


Figure 13: (a) Isometric view of one of the robot's arms and specific links. (b) CAD model of the optimized robot.

age to be given to the DC motors was 30 V. Their speed control was achieved by pulse-width modulation (PWM) through the positive and negative inputs. The final assembly with all the previous referred components, such as the arms, end effector and encasing can be seen modelled in Figure 13(b), and the built prototype in Figure 14.

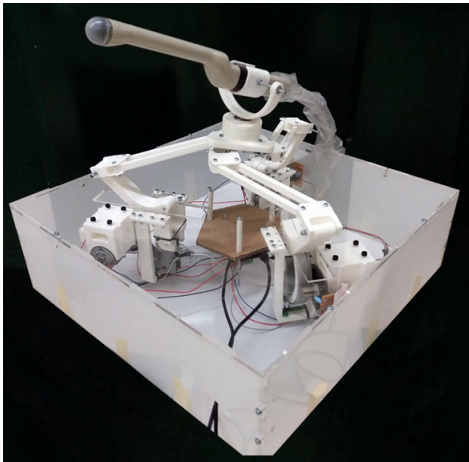


Figure 14: Prototype of the optimized robot.

### 5.3. Arduino Board

The original board was discarded and an *Arduino Uno<sup>TM</sup>* was selected. It is a low cost microcontroller with a C++ based interface. It has 14 digital input/output (I/O) pins, 6 with PWM outputs, USB connection for serial communication and a power jack for external power [9]. However, the technical specifications for the Arduino advise a maximum input voltage of 12V. For this reason, there was the need to use an adapter that allowed a greater power supply. The solution was the *DFRobot<sup>TM</sup> 2x2A DC Motor Shield*. It allows

to drive of up two DC motors with up to 2A per channel. Two shields were implemented on top of the Arduino board to control the robot's three motors.

### 5.4. Sensors

In this approach, two optical encoders were assembled for each arm. One encoder was incremental, with an encoder disk, that enabled the count of the number of angular displacements defined by each breach in the disk. The second was to be used as a homing position sensor. The measured signal for each encoder was a sinusoidal wave with the frequency proportional to the angular velocity of the motor. However, the maximum peak to peak amplitude of the wave was around 1.5V, as can be seen in the yellow signal of channel 1 in Figure 16. This is less than the sensitivity of the Arduino board to read signals as HIGH or LOW. Thus, the signal coming from the raw circuit of the encoder didn't suffice to read the number of angular displacements correctly.

#### 5.4.1 Amplifier Circuit

It was necessary to develop an amplifying circuit that would increase the peak to peak voltage of the signal, while maintaining a certain low voltage level for the LOW threshold. After some testing, the final result was the use of the high accuracy instrumentation operational amplifier (opAmp) AD620. With this amplifying circuit, it was possible to increase the peak to peak average voltage to approximately 3.3V, as can be seen in the blue signal of channel 2 in Figure 16. These values were more suitable to be read by the digital input pins, even though there are still some oscillations from motor to motor. The Arduino board was able to register all encoder rotations, which was verified even for higher rotations of the motors. The encoder circuit shown in the upper-right corner of Figure 15 was designed.

### 5.5. Serial Communication

The protocol used for communication was serial from the Arduino to MathWorks' Simulink<sup>®</sup>. It was necessary to send to the Arduino the desired velocities to each one of the motors, and the desired direction. There was also the need to receive from the Arduino board, the readings of each one of the encoders, so as to count the rotations of the motors. For this, a routine was used which provided triggering of pin change events [10]. Whenever an encoder registers one passing of a disk's slit, a RISING edge is detected and the routine checks the direction of the motor, and increments or decrements



the rotation counter, accordingly. The communication block can be seen in Figure 17.

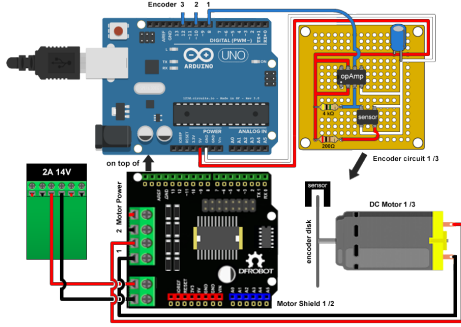


Figure 15: Electronic assembly with the Arduino, DFRobot shields, encoders, motors and the amplifying circuits.

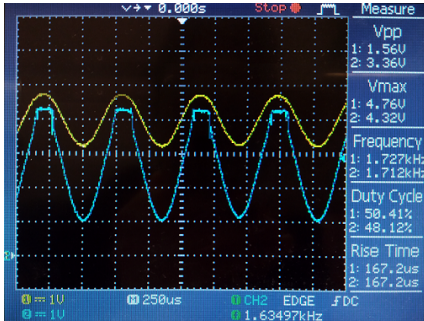


Figure 16: Oscilloscope registration of the raw signal coming from the optical encoder (channel 1 in yellow) and the amplified signal, output from the circuit (channel 2 in blue).

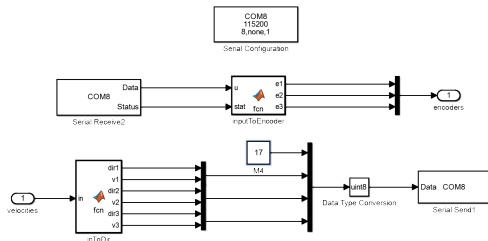


Figure 17: Simulink for serial communication.

## 6. Control Planning

### 6.1. Control Model Study

An initial feasibility control test was prepared with the original haptik device. The theory behind this approach was that a gain could control the cartesian position of the robot, since we already had access to cartesian force inputs through the *Haptik library* [11]. The further away the robot needed to move, a proportional value was transmitted as what force needed to be actuated in each direction. The control implemented is represented in Figure 18. The gain applied was determined experimentally.

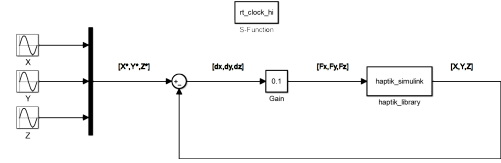


Figure 18: Control model of the original Falcon.

### 6.2. Results

As referred, the objective was to obtain a proof of concept of the control model with the original Falcon. The movement was defined as an ellipse in the xOy plane. A cosine wave was given as input to the X position, a sine wave to the Y position, and an almost null sine wave was defined for Z. Because the original workspace had an amplitude of around 60mm to each direction of x and y, a desired amplitude of 40mm was defined in this direction, while in z the amplitude was almost null. A velocity of around 0.5 rad/s was chosen which is roughly equivalent to 30 deg/s. The results are in Figure 19. The system followed reasonably well the desired position. However, there is clearly a big delay between the actual position and the desired input, which could be improved, for instance, with the use of a derivative term in the controller. Also, an integral term could be used to eliminate any steady state errors that may not be noticeable in the current output.

### 6.3. Proposed Control Model

The methodology proposed to control each one of the DC motors is presented in Figure 20. The input is a desired velocity for each one of the joints. A PID controller is to be tuned for each one of the motors. A general discrete derivative is applied so as to obtain a numerical approach to the current velocity of each joint. The block 'encodersToAngles' is the transformation from the optic encoder counter sent by the Arduino to actual joint angles, in radians. The ratio of the cable transmission system is considered as well as the corresponding angle to one slit passage of the encoder disk.

## 7. Conclusions

The robot was accurately modelled for different kinds of mechanical analysis, from statics to dynamics, and these models were implemented in Matlab. After this, an optimization was conducted in order to enlarge the workspace of the robot. With the results of the optimization process, the robot was extensively 3D modelled and analysed. A communication serial system was developed to communicate with the optical sensors and power the mo-

tors. Also, these sensors had custom made circuits to allow the board to properly count the rotations of the encoders' disks. A control feasibility assessment was presented and a future control model was proposed. This model implements custom tuned PID controllers for each motor's velocity, and also, a general position control loop that uses the Inverse Geometric Model to guide the robot through a defined cartesian trajectory.

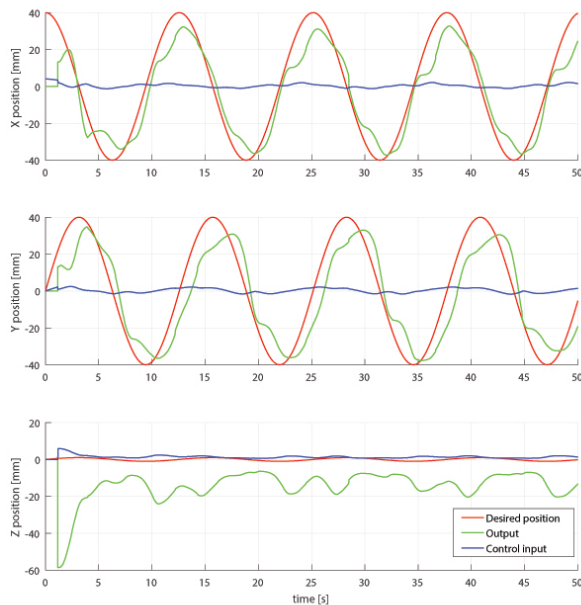


Figure 19: Results obtained with the desired positions:  $X^* = 40 \cos(0.5t)$ ,  $Y^* = 40 \sin(0.5t)$ ,  $Z^* = 0.05 \sin(0.5t)$ .

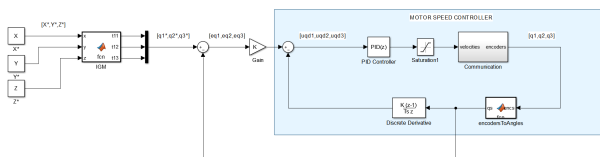


Figure 20: Proposed position control model.

## 8. Future Work

It is necessary to further investigate the possibilities revolving the control solution. Since this robot is to be used in a real-time, image feeding system from a TRUS and MRI based fusion, it is necessary to guarantee the appropriate frequency of communication. Parallel robots such as this one have been used in the fastest pick and place tasks in the world, but also in fine, precision surgeries where a very rigid and robust manipulator is needed. So, this work ends with the proposition to adapt this prototype for other kinds of surgical interventions, such as brain surgery.

## References

- [1] D. Tavares. *MRI / TRUS data fusion - Prostate cancer diagnostic and focal treatment*. PhD thesis, Instituto Superior Técnico, 2012.
- [2] M. Humbert, N. Grey, J. Muller, and C. Esling. Determination of a mean orientation from a cloud of orientations. application to electron back-scattering pattern measurements. *Journal of Applied Crystallography*, 29:662–666, 1996.
- [3] R. Stamper. A three degree of freedom parallel manipulator with only translational degrees of freedom. *Drumstage.Lib.Umd.Edu*, page 211, 1997. doi: 10.1.1.24.7361.
- [4] A. V. Shah, S. Teuscher, E. W. McClain, and J. J. Abbott. *How to Build an Inexpensive 5-DOF Haptic Device Using Two Novint Falcons*, pages 136–143. Springer Berlin Heidelberg, 2010. ISBN 978-3-642-14064-8. doi: 10.1007/978-3-642-14064-8\_21.
- [5] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo. *Robotics: Modelling, Planning and Control*. Advanced Textbooks in Control and Signal Processing. Springer, 1 edition, 2008. ISBN 978-1-84628-641-4. doi: 10.1007/978-1-84628-642-1.
- [6] T. Yoshikawa. *Foundations of Robotics: Analysis and Control*. MIT Press, Cambridge, MA, USA, 1990. ISBN 0-262-24028-9.
- [7] D. J. Block, M. B. Michelotti, and R. S. Sreenivas. Application of the novint falcon haptic device as an actuator in real-time control. *Paladyn, Journal of Behavioral Robotics*, 4(3):182–193, 2013. doi: 10.2478/pjbr-2013-0017.
- [8] K. Machulis. Everything i know about the novint falcon as of march 2008, March 2008. Available at <https://kyle.machulis.com/2008/03/25/everything-i-know-about-the-novint-falcon-as-of-march-2008/>.
- [9] A. Arduino. Arduino uno technical specs, 2017. Available at <https://www.arduino.cc/en/Main/arduinoBoardUno#techspecs>.
- [10] M. Schwager. Enableinterrupt library, 2015. Available at <https://github.com/GreyGnome/EnableInterrupt/blob/master/>.
- [11] M. D. Pascale and D. Prattichizzo. The haptik library. *IEEE Robotics Automation Magazine*, 14(4):64–75, Dec 2007. ISSN 1070-9932. doi: 10.1109/M-RA.2007.905747.