

University of Twente

EEMCS / Electrical Engineering
Control Engineering



Feedback control for steering flexible bevel-tip needles

Mitchel Bakker

BSc report

Thesis committee:

Dr. S. Misra

Msc M. Abayazid

Dr. M. Dhalle

September 2011

Report nr. 018CE2011

Control Engineering

EE-Math-CS

University of Twente

P.O.Box 217

7500 AE Enschede

The Netherlands

Contents

CONTENTS	1
1. INTRODUCTION	2
1.1 Needle insertion	2
1.2 University of Twente; Medical Robotics	2
1.3 Aim of the project	2
1.4 Outline of the report	3
2. RELATED WORK	4
3. NEEDLE BENDING MODEL	7
3.1 Circular path movement in 2D	7
3.2 Model validation	9
3.3 Conclusion on model & model validation	13
4. CONTROL SYSTEM & SIMULATIONS	14
4.1 Type of control algorithm	14
4.2 Design of the control algorithm	15
4.2.1 Steering towards a target	15
4.2.2 Inaccuracies in curvature	19
4.2.3 Estimation of the curvature	19
4.2.4 Change of curvature during insertion	24
4.2.5 Noise & Hysteresis	27
4.2.6 Steering along a path	30
4.2.7 Conclusion on Control Algorithm	30
5. EXPERIMENTAL RESULTS AND VALIDATION	31
5.1 Experimental setup	31
5.2 Experimental results	34
5.3 Proof of concept: Steering towards a target or along a path	40
5.4 Model modification	42
5.5 Conclusion on results and validation	45
6. CONCLUSION AND RECOMMENDATIONS	46
6.1 Conclusion	46
6.2 Recommendations	47
6.3 Acknowledgement	49
REFERENCES	49
APPENDICES	50
A Matlab & Simulink	50
B Elmo code	53
C Algorithms used to determine the needle curvature.	54
D Preliminary control algorithm	56

1. Introduction

1.1 Needle insertion

The trend of contemporary medicine is towards less invasive and more localized therapy. Many routine treatments employed in modern clinical practice involve percutaneous insertion of needles and catheters for biopsy and drug delivery. These applications require the needle tip to be positioned accurately and safely at a target, be it a lesion, organ or vessel [1]. Therefore robot-assisted needle insertion has attracted considerable attention for the use in procedures such as biopsies and brachytherapies, which consists of placing radioactive seeds precisely inside a (prostate) tumor. Such procedures can be quite difficult because of a number of reasons: errors are introduced when placing the needle for insertion, the needle can be deflected from its intended path by tissue inhomogeneities and the target may shift because of tissue deformation caused by the rigid needle or movement of the patient [2]. These errors can result in malignancies not being detected during biopsy, injurious pressure on tissues and increased patient trauma. A possible method to improve needle targeting is to use flexible asymmetric bevel-tip needles that are robotically steered inside the body of the patient. This method could be used in medical procedures to reach subsurface targets that are inaccessible by straight-line trajectories. Flexible bevel-tip needles deflect once they are inserted into a tissue. This deflection can be used by a robotically controlled insertion setup to avoid obstacles, such as bone or sensitive tissue. Robotically steered needles would give the physician maneuvering capabilities and precise targeting while using thinner and more flexible needles would minimize patient trauma [3].

1.2 University of Twente; Medical Robotics

At the University of Twente, the Medical Robotics (MR) group within Robotics and Mechatronics (RAM) is doing research on the application of using flexible needles in surgery to reach a certain target. So far multiple tests have been done on soft tissue simulants (gels) for detecting the amount of needle deflection, the needle tissue interaction and designing a mechanical model of needle bending.

1.3 Aim of the project

This Bachelor assignment will focus on the modeling and control part of the flexible needle project. It will be the start of designing a control system for steering of flexible needles inside a tissue. The aim of the project is to design a real-time feedback control system for steering a flexible needle during needle insertion into a soft tissue phantom. The deflection of bevel-tip needles will be used for accurate robotic steering of the needle tip towards a target or along a predefined path of sub-targets inside the tissue. For this to work, a model of the needle deflection and a control system will be designed to steer the needle guide. The objectives of this assignment can be split into three parts:

1. Modeling; to make a kinematic model of needle bending inside a tissue.
2. Control; designing a control algorithm by using this model and simulations.
3. Validation and tuning; validating and refining the model with experimental results.

Eventually, combining the feedback controller with the mechanical model and real-time visual feedback should result in a real-time control system for needle insertion that is capable of avoiding obstacles and reaching a target.

1.4 Outline of the report

The report is organized as follows. Chapter 2 describes the related work. In chapter 3 the model for needle deflection during insertion into soft tissue is described. Chapter 4 explains the needle control system and presents the simulation results. Chapter 5 shows the experimental set up and the experimental data and finally chapter 6 gives us the conclusion of this assignment and recommendations for future research on controlled needle steering.

2. Related work

Several groups have already examined the use of robotically steering flexible needles through tissue (for example Glozman 2007, Hauser 2009, Misra 2010, Webster 2006) [1] [2] [3] [4]. In most of these papers asymmetric tip (bevel-tip) needles are used. The tip asymmetry of a bevel-tip needle results in a resultant force normal to the insertion direction, see Fig 2.1. The resultant force will cause the needle to naturally bend and follow a curved path when inserted into a soft tissue, see Fig 2.2. Inside a tissue this deflection can be used for accurate robotic steering of the needle tip at the required location. This steering occurs by rotating the needle about its longitudinal axis as will be explained in the following chapters.

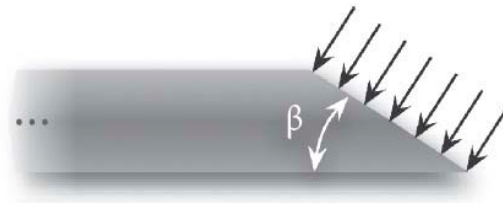


Fig 2.1. Asymmetric (bevel) needle tip with bevel angle β [3].

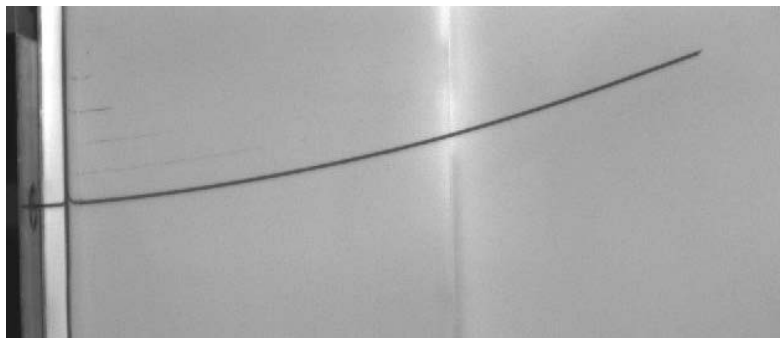


Fig 2.2. Needle deflection when inserted into a tissue simulant.

The needles that were used in these projects resemble a sort of flexible biopsy needles. A biopsy needle doesn't have a hollow tip so this results in the frontal surface area on which the resultant force of the tissue will act. The reason for this solid tip is because a biopsy needle consists of two (or more) parts, one inner and one outer layer. When a biopsy is performed and the target inside the patient's body is reached, the inner needle is pushed a little further into the desired tissue. A sample of this tissue is taken with the hollow chamber that is behind the inner needle tip, Fig 2.3. The solid tip prevents other tissue from getting into the sampling chamber during insertion.



Fig 2.3. Biopsy needle [9]

According to research by Van Veen (2011)[8] , MR group, bevel-tip needles will result in the most needle deflection compared to others with different shaped needle tips(diamond, franseen and conical shapes). For the control system to have enough steering capability as much deflection as possible is desired. Therefore in this assignment also flexible bevel-tip needles will be used for needle steering.

Misra (2010)[3] designed a mechanics-based model of robotic needle steering, which can be used to predict needle deflection based on fundamental mechanical and geometrical properties of the needle and tissue. The model uses the loads developed at the tip of the needle, the geometry of the bevel edge and the material properties of the soft tissue simulant to estimate the amount of deflection. Roesthuis(2011), MR group, developed a mechanical model that is also based on this model.

In Glozman's research (2007)[1] an image guided robotic steering system is used to guide flexible needles through soft-tissue. The guidance consisted of a real-time imaging system and an inverse kinematic algorithm made it possible to maneuver the needle by actuating the base. The needle tip followed a planned trajectory in two dimensions (2D). The control algorithm consisted of a PID (Proportional, Integral and Differential) controller which values were changed depending on the needle shape. My model is also based on a kinematic algorithm for steering the needle in 2D. It is quite comparable to steering a needle in a 2D segment of the patient's body obtained with ultrasound imaging or MRI. The main difference is that the control system will only take the motion of the needle tip into account instead of the whole needle shape. Because of this and because of the steering capabilities of the needle, the controller has not been based on a PID system as will be explained in section 4.

In line with the research of Webster et al. [4], no tissue deformation is assumed because the model is also based on thin (0.8 mm) flexible (Nitinol) needles that are inserted into a relatively stiff tissue, elaborated further in section 5. As mentioned above, the model will not take the needle shape into account but will mainly focus on the needle tip trajectory. This assumption is based on Webster 2006; he demonstrated that the needle shaft follows the trajectory of the tip almost exactly when the properties of the needle are selected appropriately with respect to the properties of the tissue through which it travels (meaning flexible needle and a relatively stiff tissue) [4]. The properties of the needle and tissue are quite comparable to the ones in this research, i.e. the use of a stiff tissue and a thin flexible needle. According to Webster the needle follows an arc with constant curvature when inserted into a stiff tissue phantom. However, this constant curvature is not observed with Nitinol needles in tissue phantoms used by Roesthuis [7]. Therefore the change in curvature during needle insertion will be determined and compared with the assumption that the needle curves with a constant radius of curvature. Substantial change in needle curvature might not occur in our experimental setup but it will most certainly occur during needle insertion into inhomogeneous human tissue. For this reason, change in bending curvature will also be included into the model and my control system will try to correct the needle trajectory for errors that might occur. The radii values have, for now, been incorporated into the model using empirical observations.

The mechanical model that is being developed by Roesthuis(2011) [7] should be able to determine the curvature of the needle and with this determine the track of the needle shaft. This is done by using force measurements at the base of the needle. In addition to this research, my assignment will focus mainly on steering the needle tip to the desired location. Once available, visual feedback (and feedback via the force measurements) will be used to determine the exact tip location and this information combined with the needle tip orientation will be used by the control system. This will be elaborated further in the next chapters.

3. Needle bending model

In this section, it is shown how an initial kinematic model for needle bending was designed in 2D. The kinematic model describes the motion of the needle without consideration of the forces that cause the motion. This should be sufficient enough since the controllers of the setup will always make sure that the predefined velocities will be kept at their set values. The needles are equipped with a bevel-tip so they will naturally deflect from their straight insertion path like mentioned in section 2. The model also assumes that needle insertion is continuous i.e. without pauses in between. The aim was to design a simplified model which could give a prediction of needle deflection when inserted into a tissue.

3.1 Circular path movement in 2D

The model is two dimensional so the needle will only bend in the (x,y)-plane. In this assignment it was initially assumed that the needle would deflect from its straight insertion path by following a circular arc, which is based upon related research (e.g. Abolhassani 2007) [6]. I also started off with a needle that is modeled to bend with a constant curvature, k . Therefore the movement of the needle tip will follow the circumference of a circle with radius R , where $R = 1/k$. The curvature of the circular path will in reality depend on multiple factors, e.g. needle density and tissue elasticity, during the insertion. The mechanical model that is designed by Roesthuis [7] will take these factors into account to determine the change in radius. However, for now, because of simplicity it is assumed that the needle bending radius to be constant.

The needle tip can follow the curvature of a circle in two ways, clockwise or anti-clockwise, depending on the direction that the bevel-tip is facing. This direction is represented by Ψ , which is the angle that determines the rotation around the needles longitudinal axis. In the model, the needle will bend anti-clockwise (upwards) for $\Psi = -90^\circ$ and clockwise (downwards) for $\Psi = 90^\circ$, so the bevel-tip can only rotate 180 degrees with respect to its initial position, fig 3.1. This results in a positive or negative angular velocity respectively. No other angles of Ψ are allowed because the needle needs to stay in the (x,y)-plane.

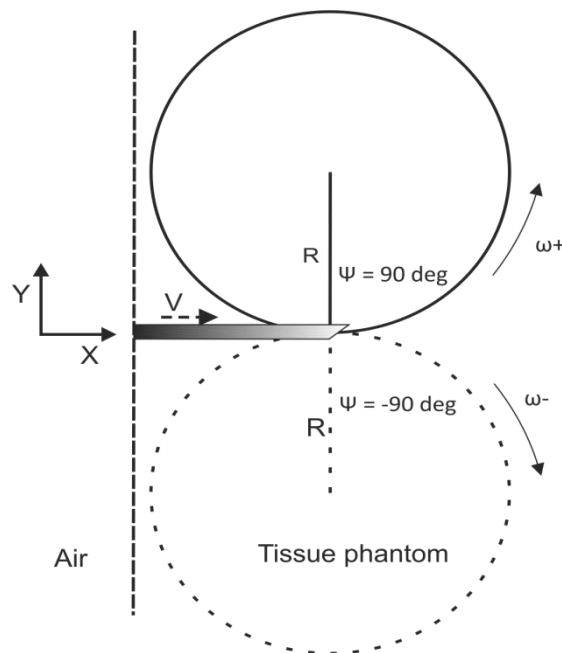


Fig 3.1.1. Needle tip orientation in 2D. Solid line shows current needle deflection curve.

The angular velocity, ω , with which the needle tip will follow the circular path, will depend on the insertion speed, v , and the radius of curvature of the circular arc according to

$$(3.1) \quad \omega = k \cdot v; \quad \omega = \frac{v}{R}$$

The current position of the needle tip is determined by integrating the velocity that was directed in the x and y direction during the insertion up to that time. The way that the insertion speed was directed in plane depends on the orientation angle (needle tip slope) between the tip of the needle and the x-axis of the reference frame which remains fixed with respect to the surroundings. This orientation angle is named theta, Θ , as shown in fig. 3.1.2.

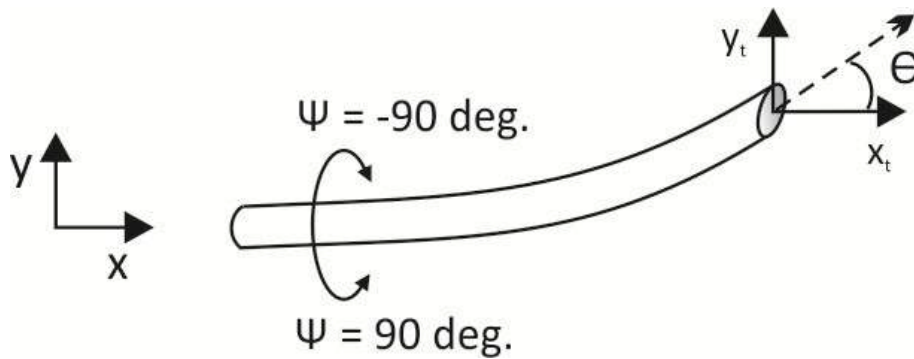


Fig 3.1.2 Needle tip position and orientation, the dotted line accounts for the direction that the needle is moving in.

The orientation angle Θ depends on the angular velocity and can be obtained by integrating ω with respect to time. Since ω is assumed to be constant, the change in Θ will be linear. V_x and V_y , the insertion speed in the x and y direction respectively can be obtained by

$$(3.2) \quad V_x = v \cdot \cos(\Theta)$$

$$(3.3) \quad V_y = v \cdot \sin(\Theta)$$

As stated before, integrating these two parameters with respect to time will give us the current needle tip location, assumed that the needle will follow the circumference of circles depending on the direction that the bevel is facing. Combining the mentioned equations gives us the Simulink model shown in Fig 3.1.3. In the following section this model will be validated.

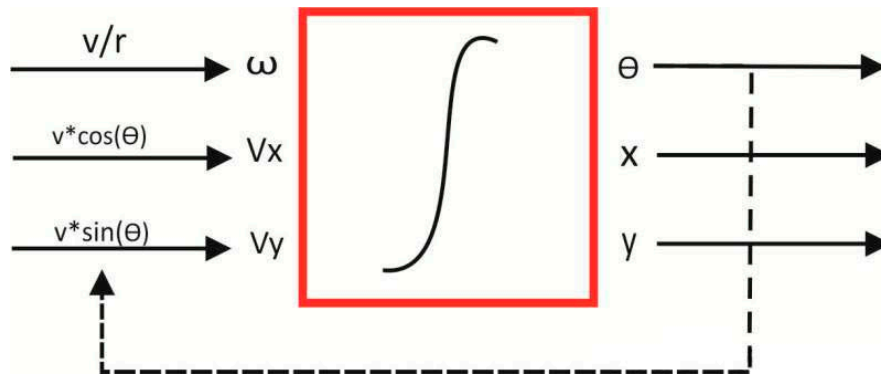


Fig 3.1.3 Needle bending model in 2D.

3.2 Model validation

The model seems plausible with the assumptions of section 3.1 which are based upon related research (Misra [3], Webster [4], Alterovitz [5], Abolhassani [6]). To validate the model empirically, single bend insertion experiments were performed and the deflection of the needle during insertion has been compared to the estimated deflection from the model. The experimental deflection data has been obtained using camera footage and the needle tracking algorithm designed by Jahya [7]. The experimental results of 11 insertion trials are shown in Fig 3.2.1. It shows the deflection during insertion of a 0.8 mm Nitinol needle into a 14.9% gelatin sample, see section 5.1. For single bend insertion the maximum deflection is 32 mm at an insertion depth of 127 mm. The maximum difference in deflection between the measurements was 3 mm. This difference is most probably caused by out of plane movement during the insertion. It was noticed from these results that the needles are not inserted perfectly straight and they already show an initial orientation angle, Θ_0 .

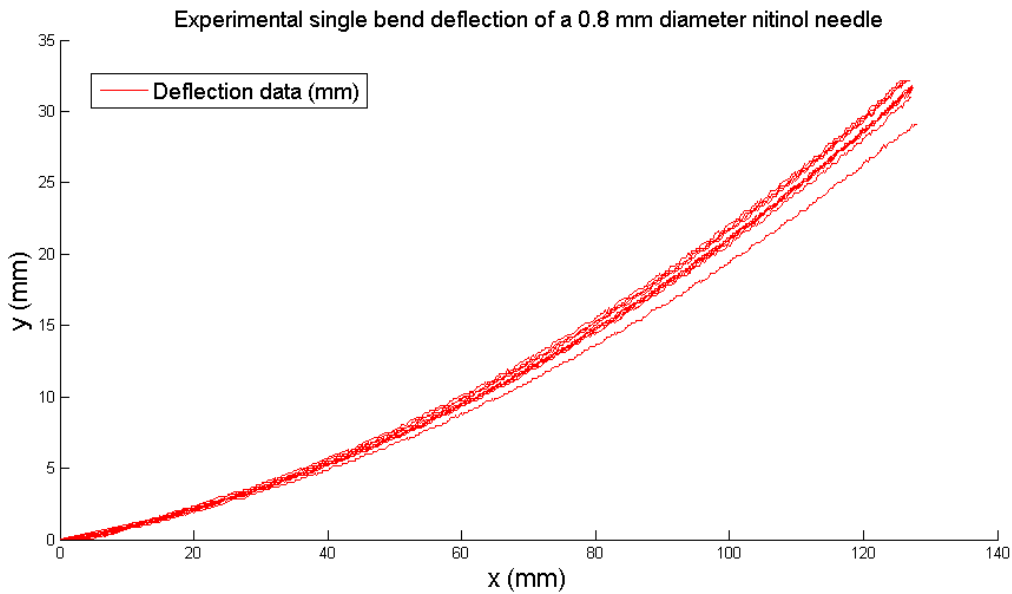


Fig 3.2.1. Experimental results of single bend deflection.

In previous research, the radius of curvature was assumed constant during needle insertion. For example, from the research of Misra et al [3], it was concluded that there is an average bending radius while a needle is bending inside a tissue. This mean bending radius only differed by an average of 8.1% from the real radii values which were determined by taking the mean of three segments of a single bend needle trajectory.

In this assignment however, it was attempted to determine if the radius of curvature changes during insertion and if adding this change to the model might show a better comparison with the real experimental deflection. It was hypothesized that the radius of curvature would increase during the insertion since the further the needle will get inserted into the tissue, the more tissue could oppose the sideways movement of the needle shaft, Fig 3.2.2.

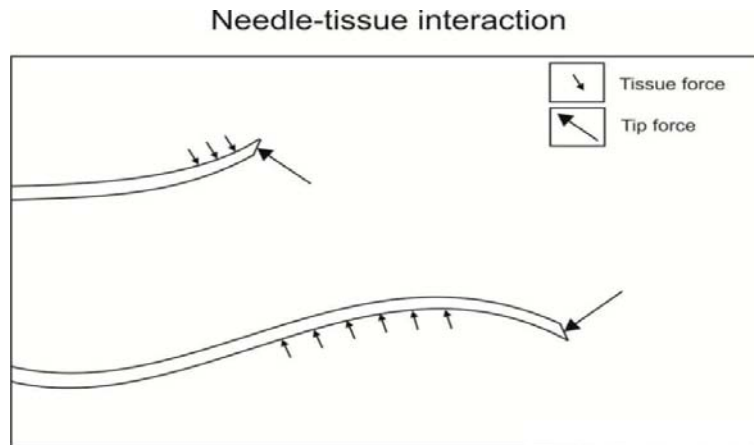


Fig 3.2.2 It was hypothesized that the tissue will resist the deflection of the needle shaft and therefore the radius of curvature would increase during insertion.

To compare the data, the estimated deflection of the model has been acquired using two different inputs: either using a constant or a variable radius of curvature as an input to the model. These radii of curvature were determined empirically from the insertion data with the use of two algorithms as shown in appendix C and described below.

1. Algorithm A:

For single bend experiments, circular arcs were fitted to the experimental data of the needle trajectory, see Appendix C. From the resulting data, the radius of curvature has been obtained which is used as an input to the model. The model assumes the needle to deflect just like the circular arc.

2. Algorithm B:

Fitting a polynomial function (third or higher degree) to the experimental data and determining the radius of curvature of this polynomial function along the insertion depth. This radius of curvature will vary along the insertion depth and it can be added as an input to needle deflection model.

The mean variable radius (Rvar) and mean constant radius (Rcon) were inserted into the model and compared with the mean of the experimental data, fig 3.2.2. The radius of curvature for insertion without rotation was chosen since this gives the most deflection and preliminary research of double bend insertion showed similar bending curves as were observed in single bend. Fig. 3.2.2 shows the deflection profile for both models using Rvar or Rcon and the average experimental deflection data with standard deviation (σ) error bars. Rvar is shown in fig. 3.3.2 and Rcon in table 3.1.

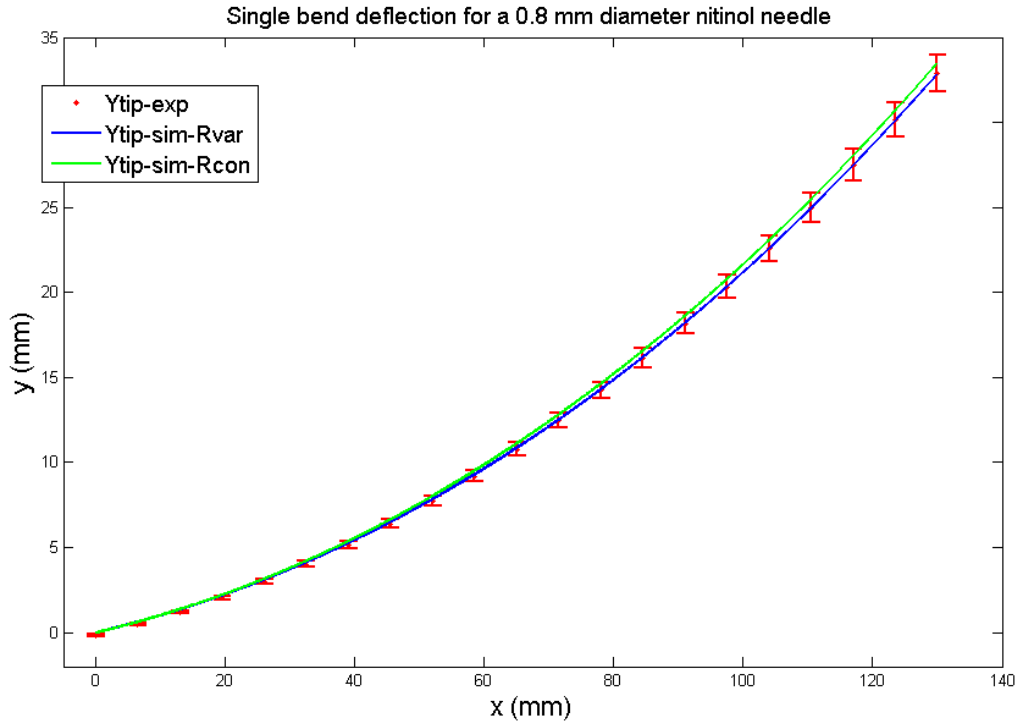


Fig 3.2.3 Deflection of the needle for single bend (no rotation), comparison of experimental (red bars), model with Rvariable(blue) and model with Rconstant(green).

Insertion	Single bend Radii (R) (mm)
#1	408
#2	417
#3	458
#4	403
#5	410
#6	443
#7	419
#8	395
#9	416
#10	420
#11	444
Mean	416
σ	19
ΔR	395 - 458

Table 3.1 Constant radius of curvature

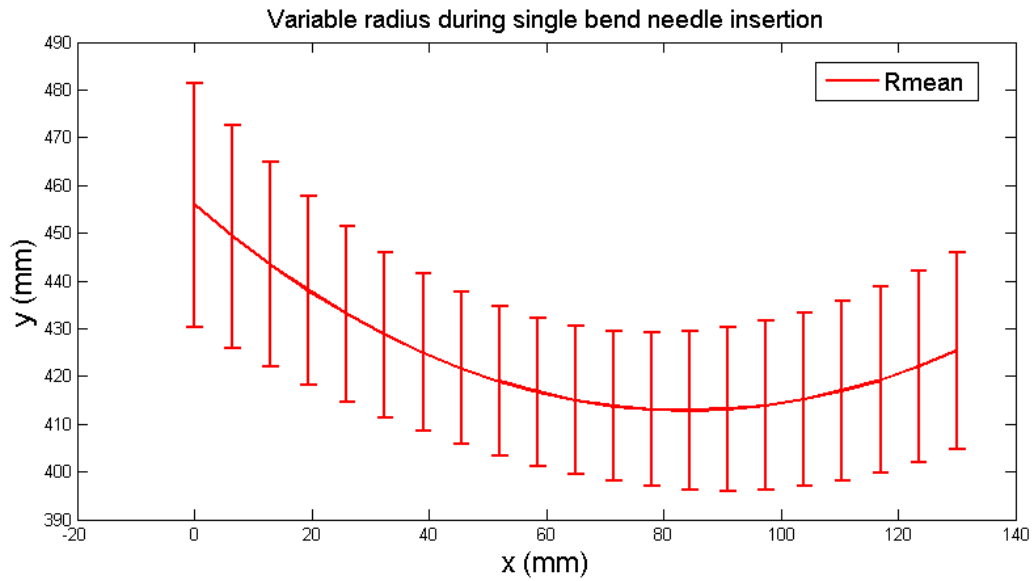


Fig 3.2.4 Mean change in radius of curvature and standard deviation (σ)

Both deflection data from the model show good comparison with the experimental data and stay within standard deviation length, σ , which varied up to 1 mm. Therefore both methods for acquiring the radius of curvature seem to be acceptable to be used as an input for the model. The results in Fig.3.2.3 showed that the radius of curvature decreases until an insertion depth of about 80 mm before increasing again and Rvar varies between 456 and 416 mm, a variation of less than 10%. Since the insertion experiments are performed in a controlled environment, in a sample with more or less homogenous tissue density, no large variations in the deflection trajectory (and therefore, the radius of curvature) were to be expected. Both radii of curvature are within close approximation to each other.

Rvar might not show big difference from Rcon in this controlled environment but it most probably will when the needle is inserted into a human body. A change in radius of curvature depends on the chosen insertion speed, tissue properties (density and elasticity), needle properties (chosen bevel-angle and needle stiffness) or to sum up: the needle-tissue interaction. A change in needle-tissue interaction might be caused by varying density of the tissue, lubrication of the needle inside the body or tissue cracks. The specific effect that all these factors may have on the radius of curvature is left for further investigation as this assignment mainly focuses on steering the needle in our test environment. It should also be noted that the model does not preclude linear paths for the needle. These can be obtained by using a large number of rotations (for 130 mm insertion, eight or more rotations).

Although the curvature changes during insertion, the assumption that the radius can be approximated with a constant considered accurate from the results in this section. For my model to be more accurate the radius of curvature should probably be updated during simulations and experiments. This might later on be done by combining it with a mechanical model for needle bending[7], but that goes beyond the scope of this assignment.

3.3 Conclusion on model & model validation

The assumption that the radius of curvature is constant in our test environment is considered valid because of the results stated in section 3.2 and the related research mentioned in this section.

Preliminary study of double bend insertion also showed similar bending curves and therefore the initial model imitated the needle movement by trying to make perfect circles as a needle trajectory.

This is what was started off with when designing the needle bending model and the control system.

The constant radius of curvature was set at the mean value of $R = 416 \text{ mm}$.

4. Control system & simulations

A control algorithm has been designed to steer the needle deflection model towards a target or along a predefined path. The needle movement in 2D has three degrees of freedom (x , y , Ψ) where y depends upon x and Ψ , fig 3.1.2. Therefore, to steer the needle, only two degrees (x , Ψ) can be used. It is quite comparable to steering a car with two fixed steering angles. The needle can be steered through the tissue and towards a target using two control inputs, the insertion (speed) and the rotation of the bevel-tip around the longitudinal-axis. I will actually only control one variable via feedback and that's the bevel-angle orientation. Van Veen (2011) has shown from his experiments that varying the insertion speed doesn't change the amount of deflection much. Moreover, from his experiments he concluded that using a small insertion speed (5 mm/s) will result in the most deflection. To steer the needle a (relatively) large amount of deflection is required so I will use this constant value for the insertion speed from now on. This chapter will focus on the design of the needle insertion control system in 2D.

4.1 Type of control algorithm

As mentioned in section 2 the controller has been designed without using PID control although it is the most commonly used feedback controller. A PID control algorithm is based upon the error between a set point (the desired needle tip position) and the value of a process parameter (the current needle tip position). Using PID control assumes the possibility of steering the needle with a variable amount of deflection depending on the sum of the integral, differential and proportional terms of the error.

However, in the 2D deflection model the needle tip deflects either upwards or downwards because of the asymmetric distribution of forces on the bevel-tip and this distribution cannot be controlled. The amount of curvature of the needle cannot be controlled; just the direction the needle tip is deflecting in. This has been done by writing a control algorithm in Matlab/Simulink code, based on the geometric properties of the assumed circular trajectory of the needle tip, see section 4.2. A preliminary algorithm is also shown in Appendix D. Steering the needle using this algorithm has however proven less effective than the control algorithm based upon geometry. It could only be used to follow a predefined path and it could not be used to plan its own course towards a target since it didn't take the position of the target into account. The geometry based control algorithm will do both. It can be used for steering towards a target by using the most optimal path and it can also be used for following a path of sub-targets as will be shown in section 4.2.

4.2 Design of the control algorithm

Steering the needle can be done by considering the geometry of the needle deflection during insertion. The control algorithm is also based upon the assumed deflection of the needle along the circumference of a circle, like mentioned in section 3.1. By using the shape of circular paths to reach a target, the rotation points and an optimal needle trajectory are determined while the needle is being inserted. The optimal path will use as few rotations as possible since rotating the needle might inflict tissue damage and this will cause patient discomfort [8]. In this section, design of a control algorithm to reach a single target or to follow a path is further elaborated.

4.2.1 Steering towards a target

At this stage of control design the needle was assumed to bend with a constant radius of curvature. I started off by defining two circles C_r (circle of rotation), C_t ('touching circle') and both their radii R_r and R_t , where $R_r = R_t = R$, the radius of curvature, see fig 4.2.1. The circumference of the first circle, C_r , is the path on which the needle tip is currently moving. The touching circle will move along with the needle tip and its circumference will stay tangent to the needle tip direction. This means that the needle tip is located precisely between the centre points of the circles. The circumference of C_t resembles the needle trajectory if the needle was to be rotated 180 degrees along its longitudinal axis at that point during the insertion.

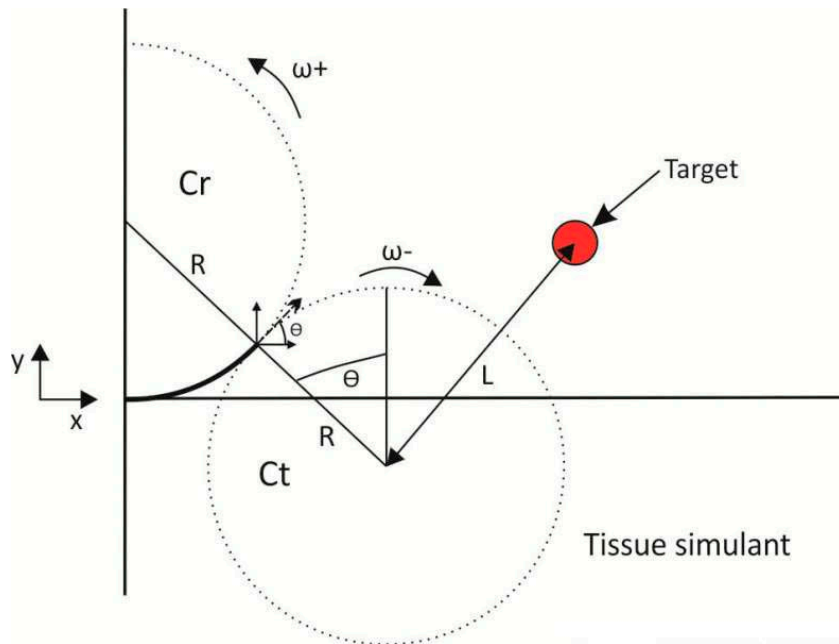


Fig.4.2.1 Needle trajectory towards a target. L is the distance from the centre of C_t towards the target. By comparing L with R it is determined if the target is upon the circumference of C_t .

With constant radius of curvature during insertion, the needle can reach a target by using just one rotation, see fig. 4.2.2. The touching circle will move along the initial needle path while the needle is being inserted. Both the needle tip and the target will be upon the same curve, once the circumference of the touching circle reaches the target. This will be the point, (1) in fig 4.3.2, on which the needle has to be rotated along its longitudinal axis. Therefore C_t and C_r switch places and the needle will rotate along the new circular arc. If the radius of this arc remains constant the needle will precisely reach the target. This steering method is applicable because the radius of curvature of the needle is big compared to the insertion depth at which the target is located.

The needle is inserted a maximum depth of 130 mm while the radius of curvature lies within the range of 350-500 mm. The insertion depth of 130 mm has been chosen because a target inside the human body, for instance in the prostate, can also be located at an insertion depth of about 100 to 140 mm [12].

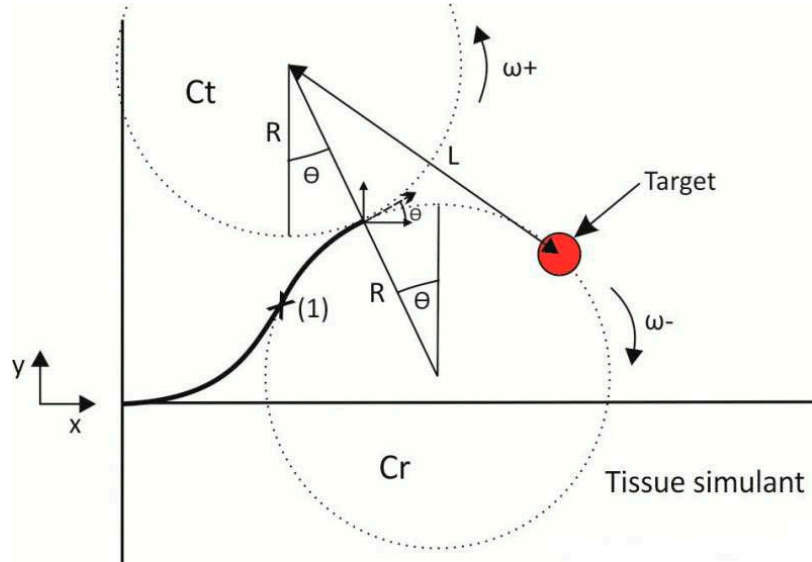


Fig 4.2.2 Needle trajectory after rotation at (1)

The flowchart in fig 4.2.3 shows the algorithm that is performed every time step ($\Delta T = 0.01$ sec). The algorithm determines the exact point of rotation by comparing the distance, L , from the centre of C_t to the target, with the radius of curvature R . If the condition $L = R$ is satisfied, the needle will rotate, see also fig 4.2.2. The target is defined as a little circular object with radius R_{target} . The control system aims to get the needle point to precisely the centre of this object, which I name throughout this report as the actual target. The algorithm stops the insertion when the target has been reached up to 0.5 mm or, when this cannot be guaranteed, the algorithm will stop the insertion when X_{tip} , x-coordinate of the needle tip, passes by X_{target} , x-coordinate of the target.

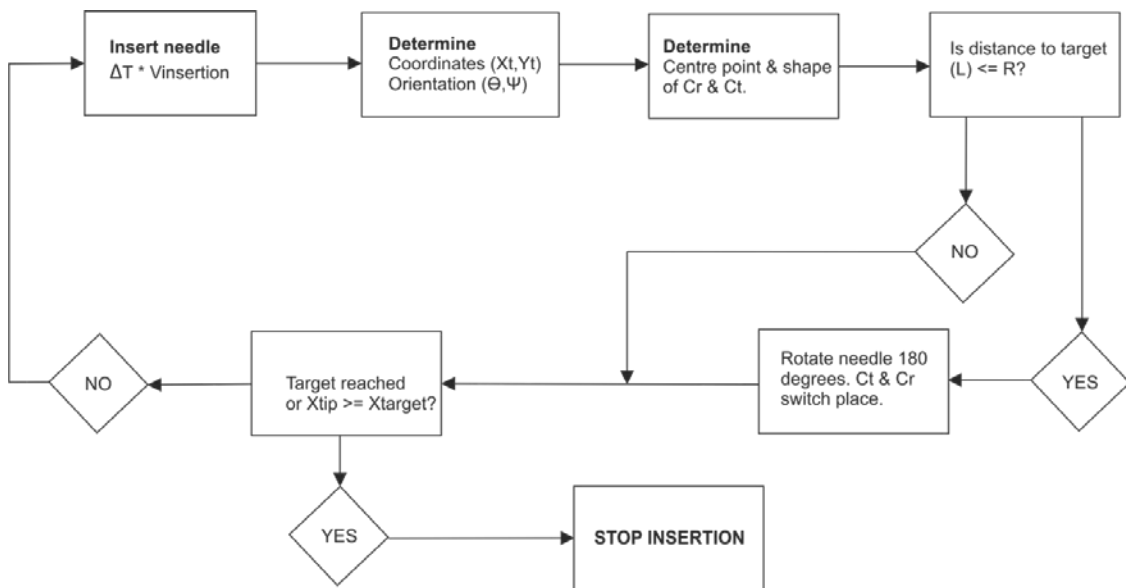


Fig. 4.2.3 Flowchart of needle insertion. The steps in the flowchart are performed every time step.

Simulation results of the needle tip trajectory are shown below. Fig 4.2.4(A) shows the trajectory of a needle with constant curvature radius $R = 416$ mm. The needle tip is steered towards a target at (130,15), this target location will be the same throughout the simulations shown in this section. Θ_0 is set to 0.06 rad. Fig 4.2.4(B) shows the direction of the bevel face and fig 4.2.4(C) shows L versus R and the point of rotation, i.e. the rotation condition: $L = R$. Once a rotation has taken place, L increases slightly because C_r and C_t switch place. The needle won't rotate again once the needle tip reaches the target.

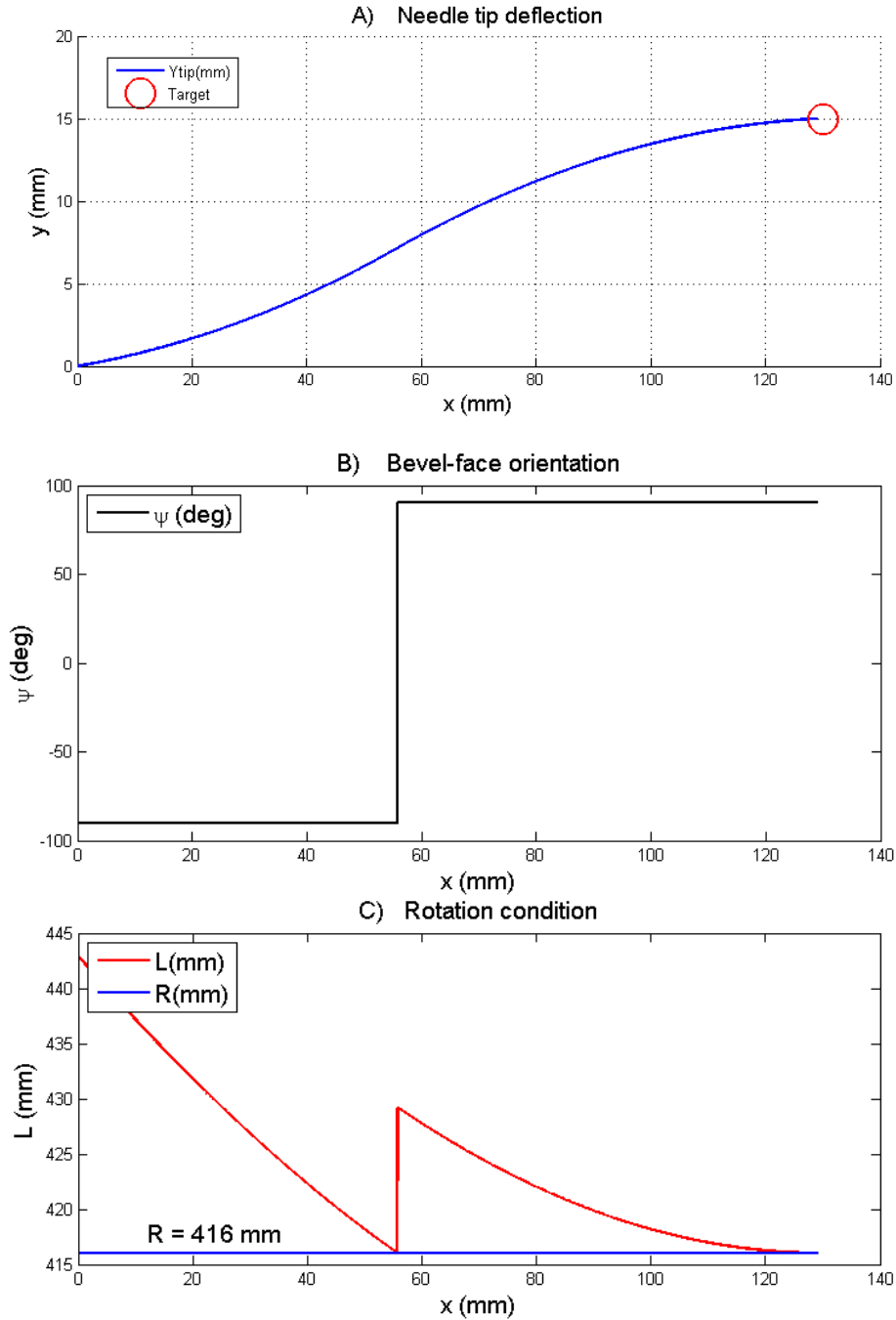


Fig 4.2.4 Simulation results of needle steering with constant radius, $R = 416$ mm. (A) Needle trajectory. (B) ψ (deg). (C) Rotation condition: L vs R .

It should be noted that in the needle rotation figures, e.g. fig. 4.2.4., it is shown that the model considers the 180 degrees of rotation of the needle to occur instantly. Of course this is not the case in the real set up, but according to the motor specifications and the power supply via the Elmo controllers, section 5.1, the motor will rotate with a frequency of 5Hz. This is 0.2 sec for a full rotation, so 0.1 sec for 180 degrees of rotation. This rotation time might increase slightly when rotation occurs inside a tissue because of inertia of the needle and friction with the tissue. However, from analyzing camera insertion data, there has been concluded that a 180 degrees needle rotation about its longitudinal axis takes about 3 to 4 frames while the camera is recording 30 frames per second (section 5). Therefore a rotation time estimate of about 0.1 – 0.13 sec seems quite accurate. During this rotation the bevel-face of the tip will also point in the out-of-plane direction but since the needle is only inserted with 5 mm/s, the out-of-plane deflection is negligible. Therefore instant rotation of the needle tip is assumed to be correct.

4.2.2 Inaccuracies in curvature

As shown in fig 4.2.4, the needle could reach a target using one rotation if, in the ideal situation, the radius remains constant. However, the needle might miss the target if the curvature of the needle during insertion is different than was expected. First of all, the target may not be reached accurately if the radius of curvature is different than initially estimated. As will be shown from experimental results in section 5.2, the assumed constant and variable radius of curvature both differed per insertion even though a tissue phantom is used with (more or less) homogeneous properties. This resulted in a deflection difference between the needle insertions that went up to 3 mm for single bend and 2 mm for double bend insertions, section 5.2. This might seem like a small error at a deflection of about 32 or 17 mm but since the size of the targets within the human body are also in the order of millimeters (4mm or larger [11]) such an error won't be acceptable. It could mean the difference between the right and wrong diagnosis of a tissue sample.

Secondly, this error in deflection occurred in our controlled test environment, the error in a real human body might be way bigger because of the inhomogeneity of the patient's tissue. This inhomogeneity might cause a change in radius of curvature during the insertion procedure. A decrease in the radius of curvature would improve the steerability of the needle whereas an increase in the radius of curvature would make steering the needle a lot harder. Practically it means that if the radius increases during insertion, the needle trajectory will gradually start to resemble a straight line and this decreases our steering capabilities. More inaccuracies might arise when, in reality, a patient moves during the insertion procedure. However, if the patient is sedated this should not be an issue and therefore this form of inaccuracy is neglected.

4.2.3 Estimation of the curvature

As shown in table 3.1, the radius of curvature will not be equal for every needle insertion. Therefore the radius of curvature on which the control algorithm bases its decisions might differ from reality. This will only become a problem when the radius of curvature, R , is bigger than the estimated radius of curvature, R_c , which is used by the controller. Simulation results of such an error are shown in fig 4.2.5. In both cases $R_c = 416$, the mean radius from single bend insertion. For the blue lined insertion $R = 458$ mm, largest measured radius from table 3.1 and for the orange lined insertion $R = 500$ mm. The target is yet again located at (130,15).

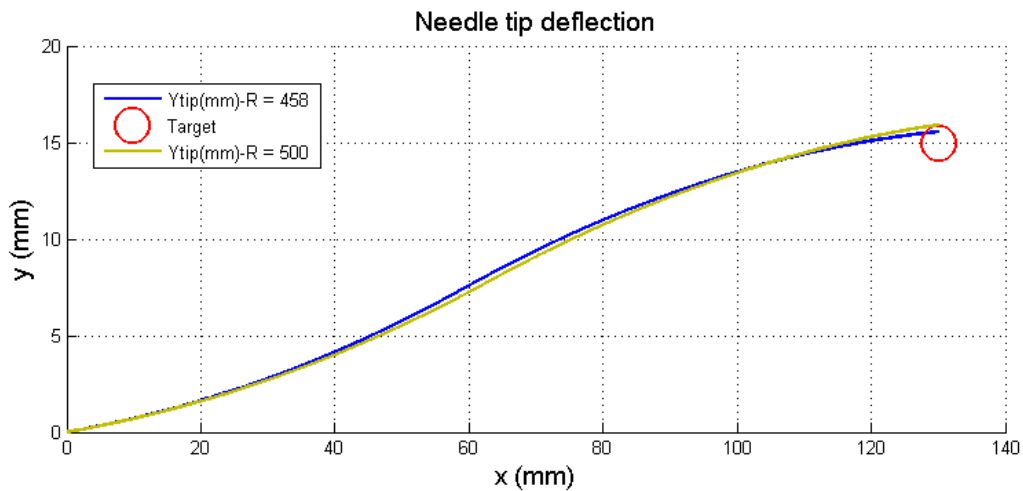


Fig 4.2.5 Needle steering towards a target when $R > R_c$.

The resulting deflection error from the centre of the target is 0.6 mm for $R = 458$ mm and it increases to about 1mm for $R = 500$ mm. For accurate needle steering such errors cannot be neglected and therefore my control system will have to take this into account.

To make sure that a target can always be precisely reached the controller should prevent a target from entering the unreachable or 'dark' area within C_r and C_t , see fig 4.2.6. In other words, the target should be kept within the 'funnel' that is defined by the curvature trajectories of C_r and C_t .

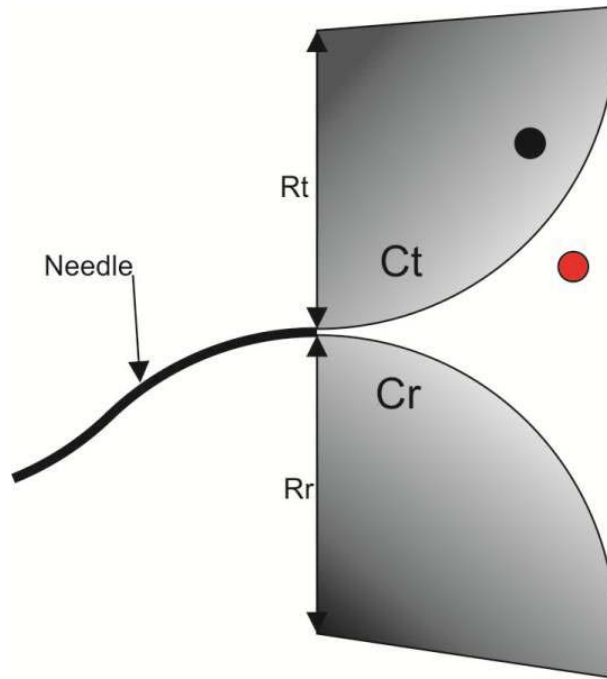


Fig 4.2.6 Unreachable area for the needle tip shown in grey; black target cannot be reached.

Taking the uncertainty in radii values into account, the controller will have to make steering decisions based on a 'worst case scenario' and makes sure that the needle can still reach the target with the largest radius (or smallest curvature) that has been observed from experimental data. The controller will make rotation decisions based upon a radius of curvature, R_c , which is at least larger than the ones observed from experimental data. In this way it will prevent a target from entering the unreachable area that is defined by the actual radius of curvature, R , see fig 4.2.7. R_c will always be kept bigger than the actual radius during insertion which may not be known since the radii values acquired from the experimental data show variation between insertions, section 5.

Both R and R_c are tangent to the needle tip and thereby define two 2D funnels with error margin, N , in between, fig 4.2.8. The error margin is created by the difference in their radii and N converges to zero as the needle approaches a target, see Fig 4.2.8 where $N_{i+1} < N_i$. The error margin will vary during the insertion since R may not be constant. With every next rotation the needle will be steered more accurately towards the target because the error margin converges to zero. The algorithm will rotate the needle when a target hits one of the sides of the funnel defined by R_c . The needle will actually deflect along the steeper curve of the outer funnel, defined by R . Therefore the needle has to be rotated multiple times since every time a rotation decision is made, the needle will actually rotate too early. It is a trade-off between staying in range to reach the target and making multiple rotations to do so.

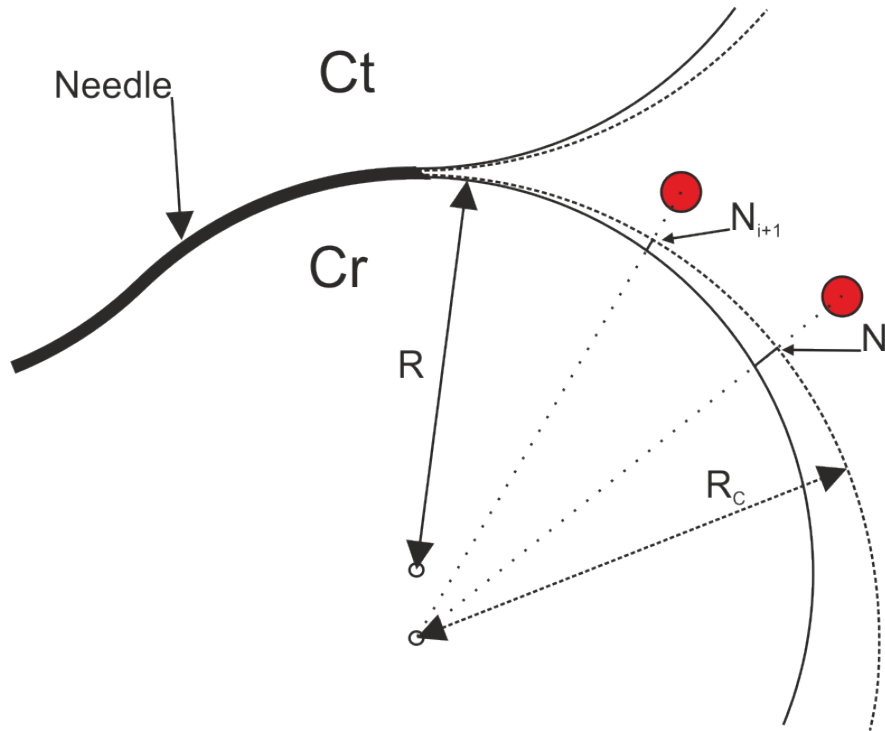


Fig 4.2.8 Needle steering using R & R_c . N converges to zero as the needle tip comes closer towards the target, the relative position of the target to the needle tip has changed from i to $i+1$.

In the Simulink simulations, the model assumes the needle to bend with the mean radius of the experimental data; it is an estimate of the radius with which the needle will most likely bend, $R = 416$ mm. R_c is set at 460, just larger than the highest R value from the single bend experimental data. Simulation results of a needle steering towards a target using multiple rotations are shown in fig 4.2.9. The effect of increasing of R_c on the needle trajectory is shown in fig 4.2.10.

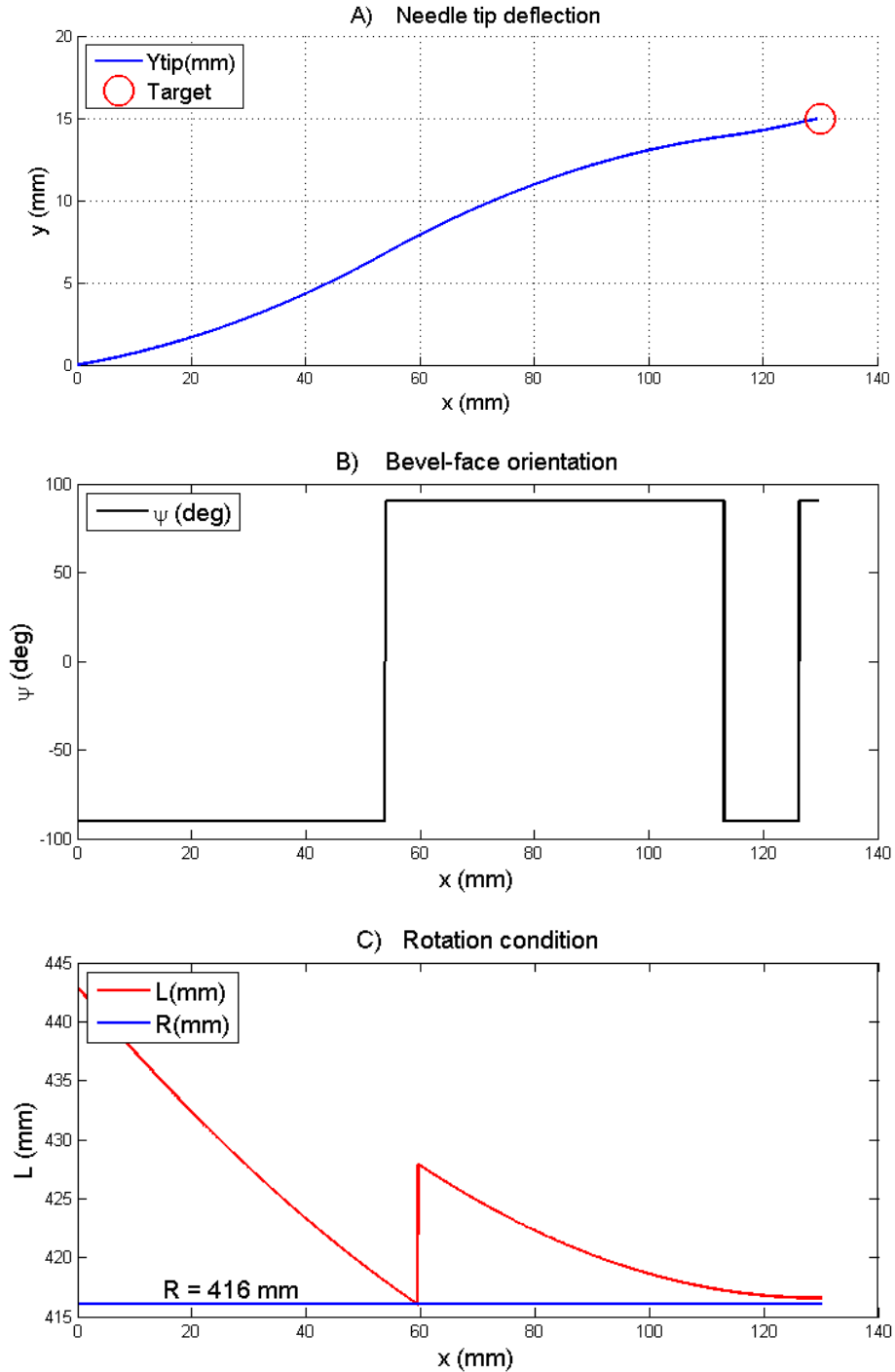


Fig 4.2.9 Needle steering for constant radius using R_c .

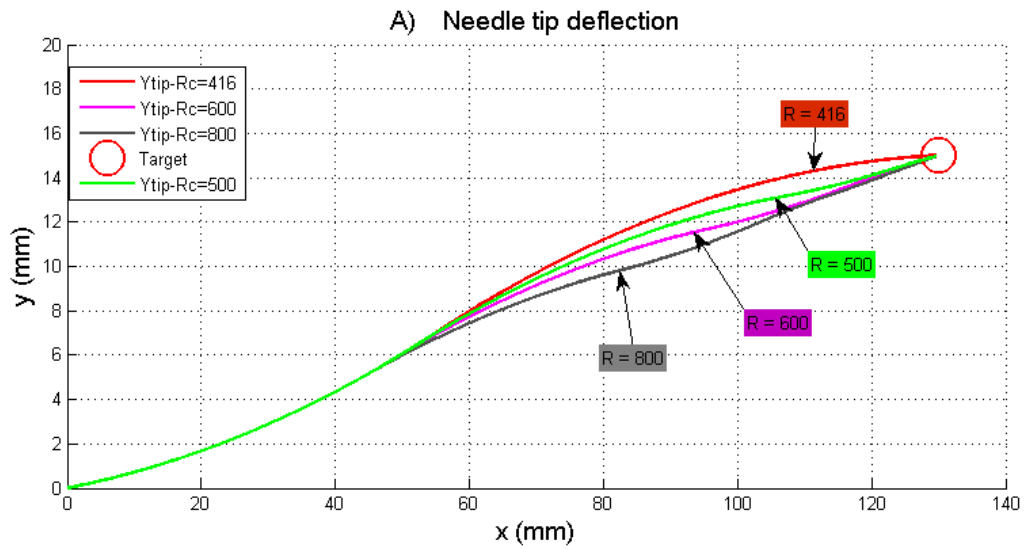


Fig 4.2.10 The effect of increasing R_c on needle tip trajectory.

When increasing R_c , the needle trajectory will start to resemble a straight line because more and more rotations take place. This occurs because the funnel defined by R_c becomes smaller. The range of targets that can be reached (deflection along the Y-axis) is still determined by the actual radius of curvature.

4.2.4 Change of curvature during insertion

As shown in section 3.2, the actual curvature of the needle trajectory might change during insertion. From the experimental data no considerable change in needle curvature is observed but this might occur if the sample tissue is more inhomogeneous. In this case the needle bending trajectory might not be comparable with simple circular arcs. Therefore also simulations have been performed where a variable radius of curvature, R_{var} , is inputted into the model. R_{var} is shown in fig 4.2.11.

Rotation decision will still be based upon R_c , in this case $R_c = 570$ mm. The variation in R between R_c and R_{var} during the insertion also leads to a variation in the error margin, N . However, this won't pose a problem in needle steering as long as the funnel defined by R_c is smaller than the one defined by R_{var} . Simulation results of needle steering with R_{var} , fig 4.2.11, and $R_c = 570$ mm is shown in fig. 4.2.12.

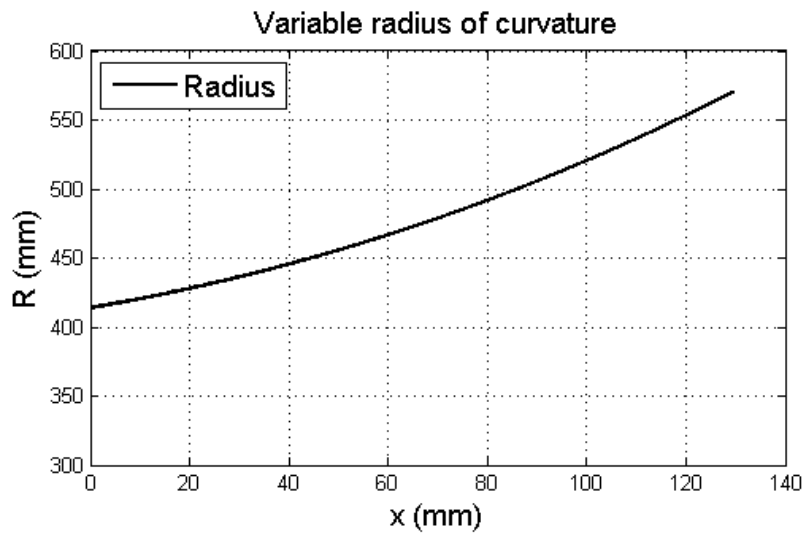


Fig. 4.2.11 Increasing radius of curvature during insertion.

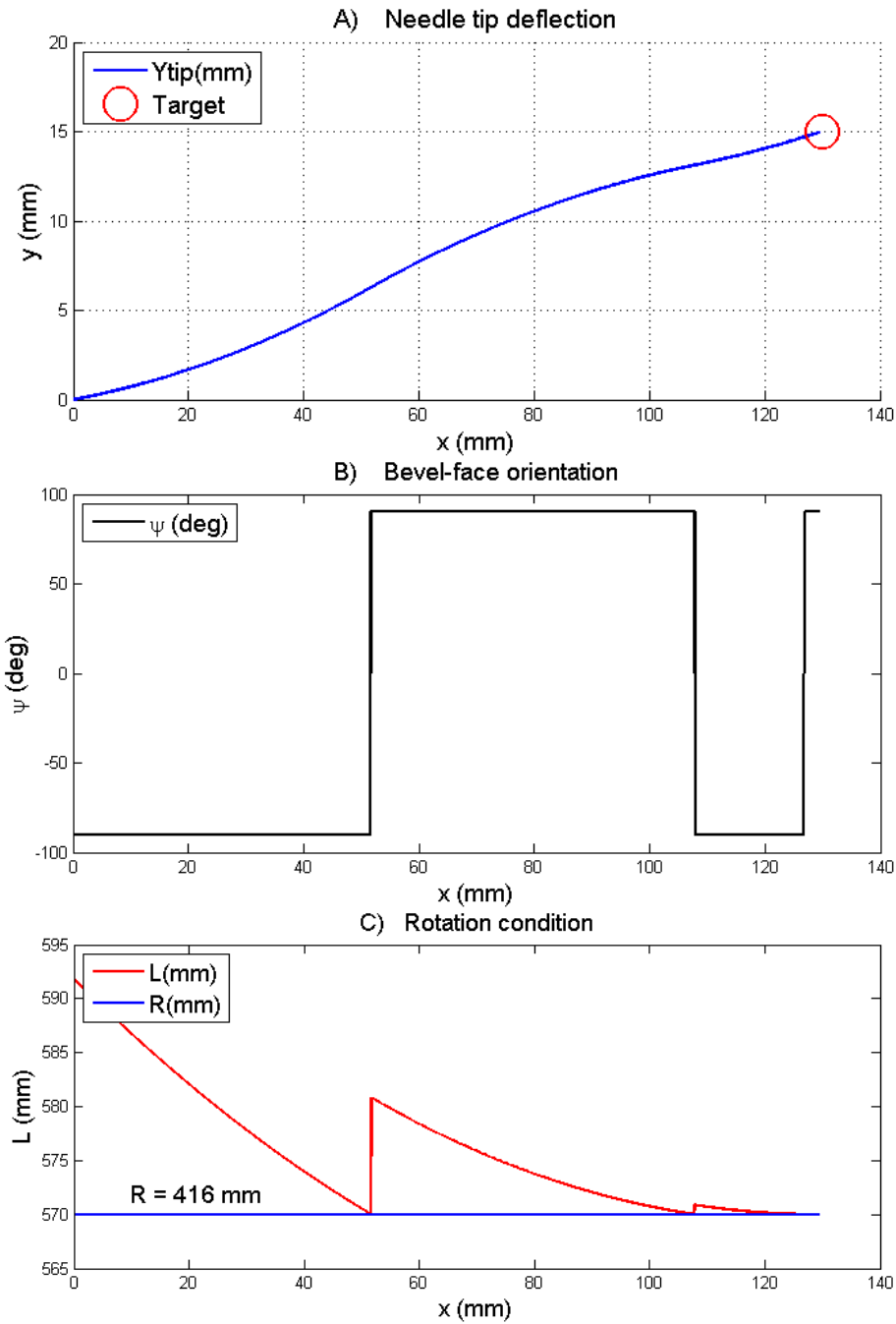


Fig 4.2.12 Needle steering, radius of curvature is variable.

As shown in Fig 4.2.12, again multiple rotations were made. This occurs because R_c is bigger during the whole insertion and therefore the funnel R_c will always be smaller than funnel R . A better approximation of the funnel R_c could be made by taking into account all future values for the radius of curvature. Therefore the funnel that is used for steering the needle shouldn't be based upon perfect circular shapes but upon a polynomial function which is a tangent to all the future circles with changing radii, see fig 4.2.13 This polynomial function will define the new funnel in which the target should be kept. In this way also the error margin between funnel R and funnel R_c could be controlled.

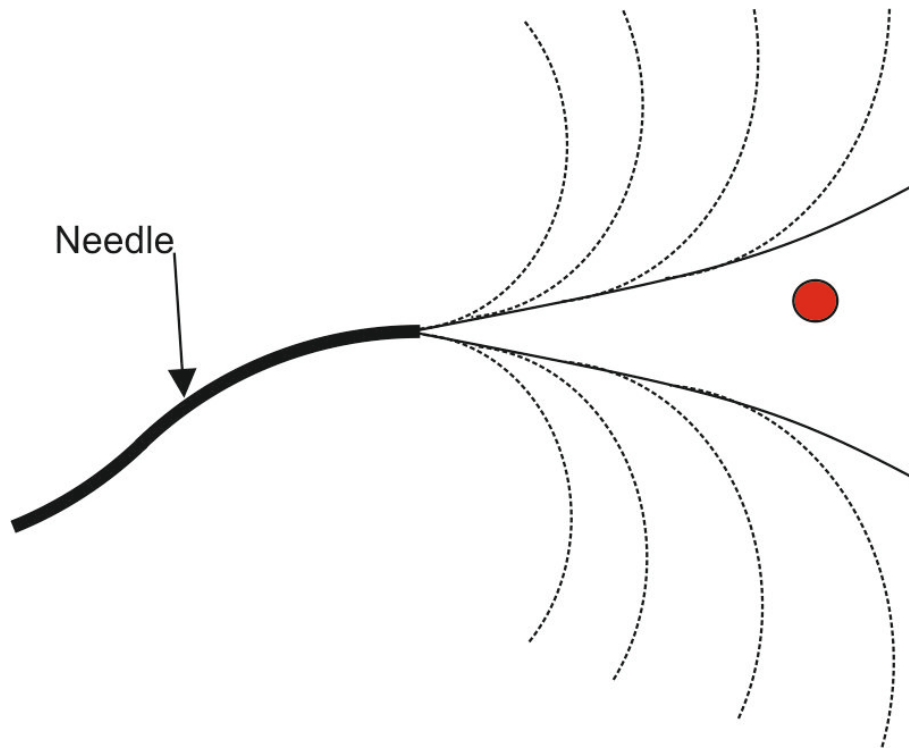


Fig. 4.2.13 Steering funnel using future radii values.

Taking into account the change in radius of curvature is yet to be implemented and is therefore left as a recommendation. In the meanwhile, the constant R_c will be used since it makes sure that a target can accurately be reached according to the simulations. As long as R_c is chosen larger than R during insertion, it will make sure that the funnel defined by R_c will be smaller than the one defined by the actual radius of curvature, R , and therefore the needle can be steered in a robust way.

4.2.5 Noise & Hysteresis

The needle tracking algorithm that was used for analyzing the video data hasn't been able to give real-time visual feedback. However, the control system needs to be capable of using the input signals that it will receive from a real-time needle tracking algorithm. The visual feedback will always include some form of noise. Alex Jahya [7], MR group, estimated the maximum amount of noise to be 1 pixel or 0.1695mm. Therefore a uniform random (noise) signal with 0.1695mm noise amplitude has been added to the position input of the controller in the Simulink model, see appendix A2. Simulation results are shown below in fig 4.2.14 and fig 4.2.15. The target is again located at (130,15).

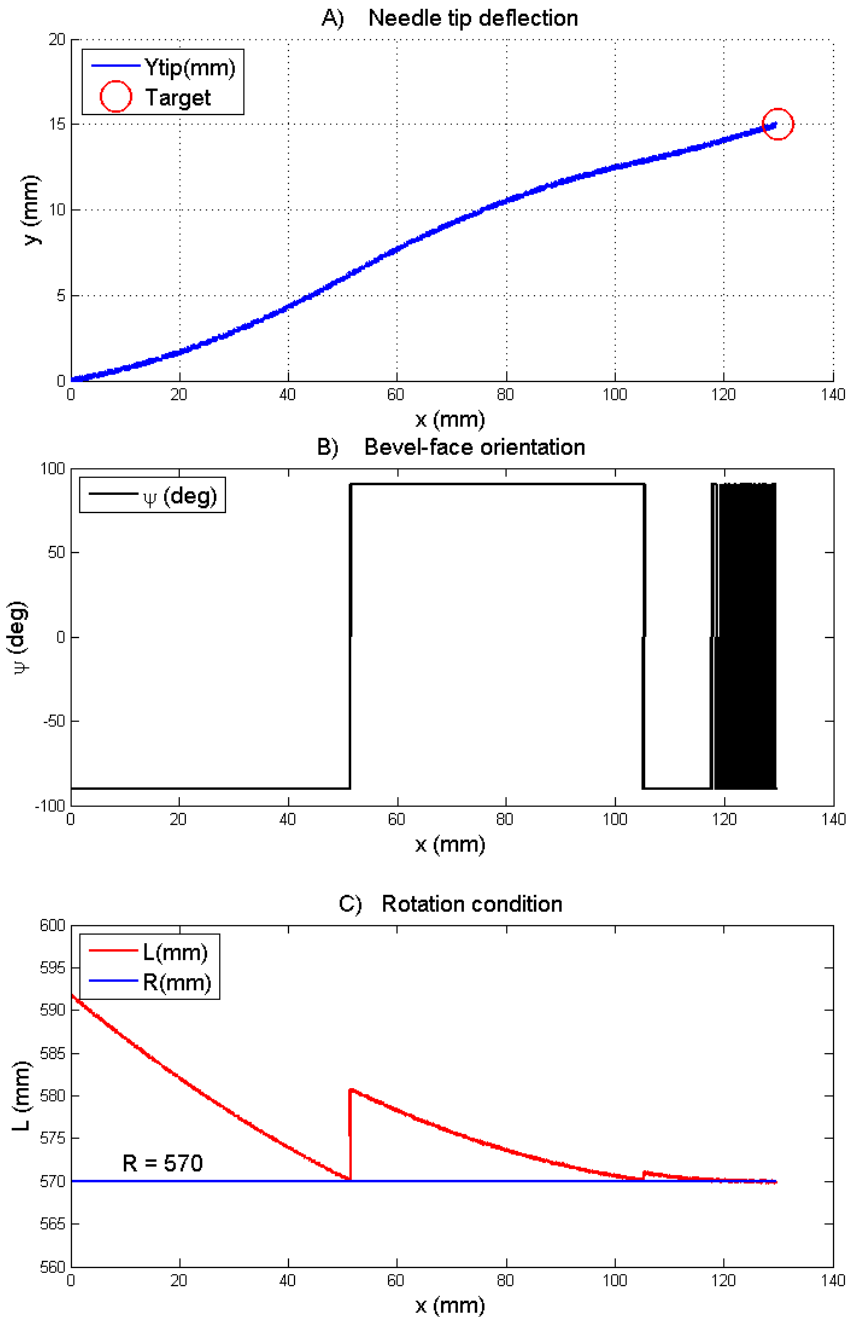


Fig 4.2.14 Simulation of needle steering with added noise. (a) Needle trajectory. (b) Ψ (deg). (c) Rotation condition: L vs R.

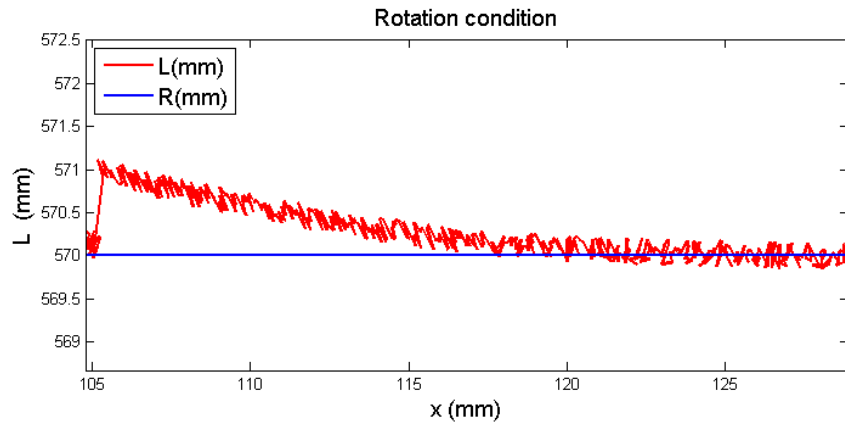


Fig 4.2.15 Zoomed image showing the effect of noise the rotation condition.

In these figures it is shown that the system will become unstable once the needle tip has almost reached the target. Because of the additive noise the worst case curvature will constantly reach the target and the controller will keep the needle rotating 180 degrees back and forth or in other words, switching between C_t and C_r . You could compare this with dropping a small ball in a funnel. If the funnel is constantly moved sideways than the ball will keep on bouncing back and forth until it finally leaves the funnel, fig 4.2.15. In this case the 2D funnel consists of the two curves with R_c , fig 4.2.8.

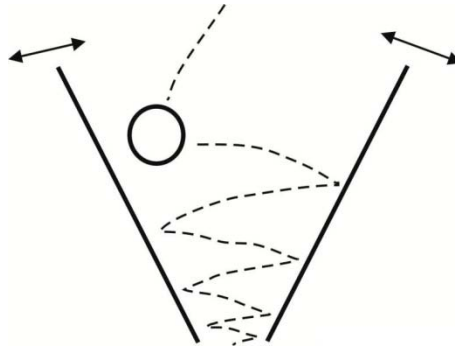


Fig. 4.3.15 Funnel + ball.

In the simulation the needle is still able to reach the target because I have assumed needle rotation to occur instantaneous. Since, in real life, my motor doesn't have the unlimited speed to achieve this, a tradeoff will have to be made by applying some form of hysteresis.

In the controller, hysteresis has been implemented by defining a bigger target, a circle with radius R_h , Fig 4.2.16, the needle will only be rotated 180 degrees once the target R_h completely leaves the funnel. R_h is defined just bigger than the noise amplitude, $R_h = 0.2$ mm, and therefore the noise will have no more influence on the steering process. As mentioned, it will be trade off, you will prevent the system from going instable but a small error in positioning the needle tip will be added, see Fig 4.2.16. The target is still accurately reached as long as R_h is smaller than R_t . Therefore also the noise in tracking the needle tip should be kept small in comparison to the size of the target.

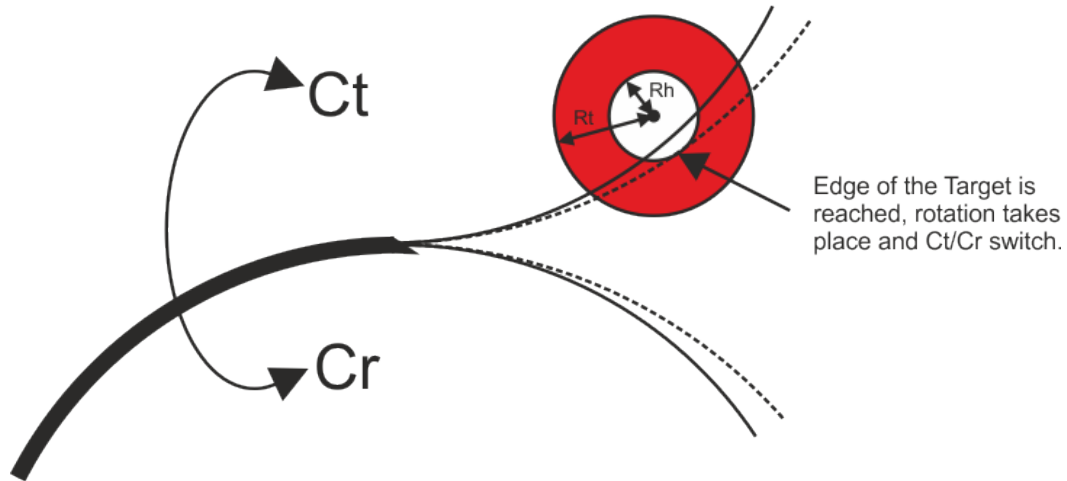


Fig 4.2.16 Needle steering with hysteresis. R_t = target radius, R_h = hysteresis radius. Solid line: needle deflection trajectory. Dotted line: decision based funnel.

4.2.6 Steering along a path

A path towards the final target will be required when the system is to make curves around obstacles and for this a path planner should be designed. Currently there is no path planner implemented yet so the paths that are defined for the needle tip to follow are either linear or polynomial paths (or a combination of both) which consists of sub-targets. The locations of these sub-targets in the 2D-plane are predefined and inputted into the control algorithm. The algorithm aims for a sub-target and once this is nearly reached (up to 0.5mm) or passed by, it will aim for the next one.

The linear and polynomial paths are not optimal paths for the needle to deflect along and therefore multiple rotations have to take place. Simulation results of the needle tip following a linear and a polynomial path are shown in fig 4.3.17. $R_c = 500$ and $R = 416$ have been used as an input to the model in these simulations.

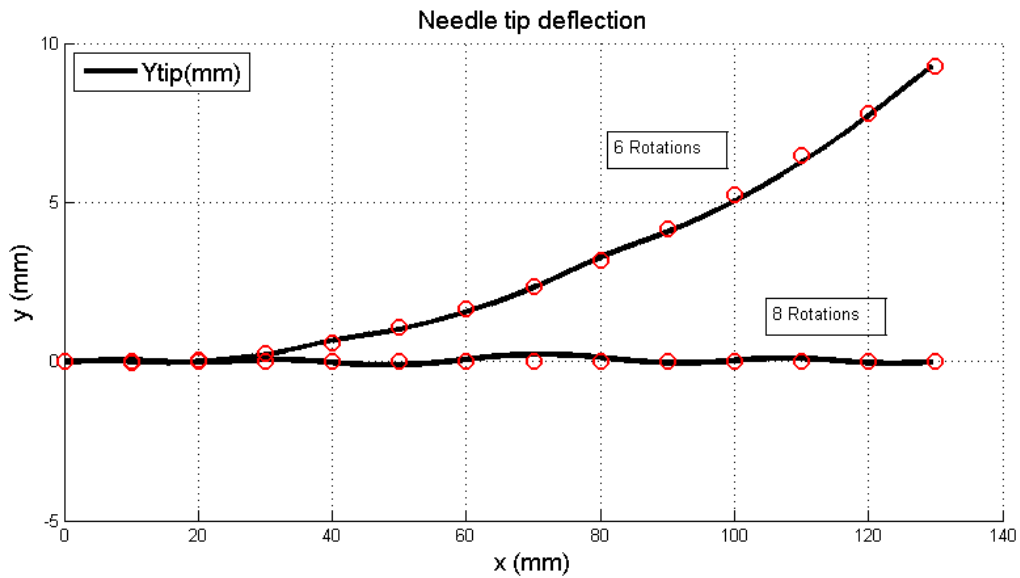


Fig 4.3.17 Needle steering along a path. 8 rotations where necessary to follow the linear path, 6 for the polynomial path.

The needle makes a sinusoidal movement around the predefined path by making in this case either 6 or 8 rotations, see Fig 4.3.17. The deflection from the path fluctuates but remains smaller than 0.25 mm. This fluctuation is caused by the implementation of hysteresis and the fact that the needle tip is always steered towards the next target without taking the following targets into account.

4.2.7 Conclusion on Control Algorithm

With the chosen control algorithm, it should be possible to steer a needle towards a target or along a path defined by sub-targets. It must be noted that the algorithm is based upon the model, section 3, which assumes some simplification on needle bending. At rotation some changes in needle bending might arise and the control algorithm has to be robust for these uncertainties. The needle tip deflection model and the control algorithm might have to be adapted when the actual needle deflection behavior is different from the simulations. This comparison shall be made in section 5.

5. Experimental results and validation

To validate my model and control system, needle insertion experiments were performed by using a two degree-of-freedom (DOF) device for insertion and rotation [7]. The needle insertion setup and experimental results are described in this chapter.

5.1 Experimental setup

The needle insertion device is shown in fig 5.1.1. It allows translation along and rotation about the needle's longitudinal axis.

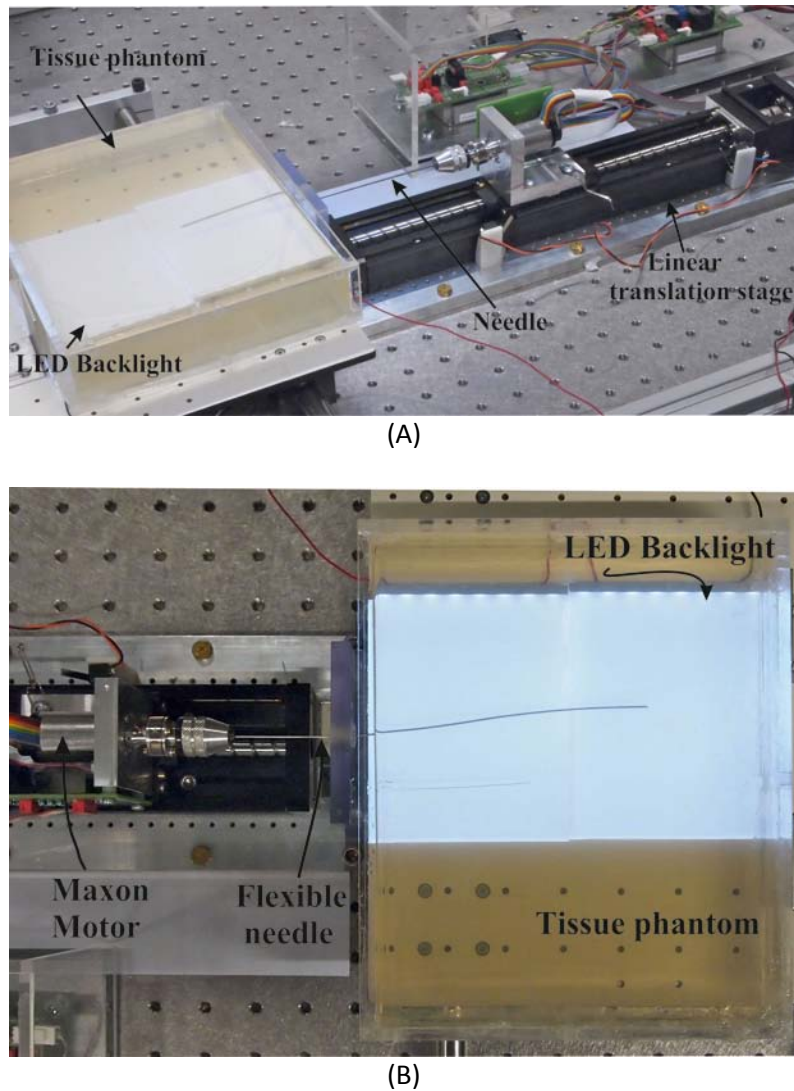


Fig 5.1.1 Two DOF setup used for inserting needles into a tissue phantom. (A) Overview (B) Topview.

The setup consists of a translation stage and a rotational setup. Both parts are actuated by a Maxon Motor, translational motion with a type RE25, with GP26B gearhead, transmission ratio 4.4:1 and rotation with type EMax22, Maxon Motor. The motors are being controlled by using position control on two Elmo Whistle 2.5/60 controllers. Besides their internal code, written in Elmo studio, the controllers interacted with the insertion procedure code that was written in Matlab, version 2010b.

Recording of the needle insertion is done at 30fps by a Sony XCD-SX90 charge-coupled device (CCD) FireWire camera that is mounted at a height of 450mm from the top surface of the gelatin phantom. Initially the setup was lighted by using a light emitting diode (LED) beam directed sideways into the tissue sample. The results from these insertion trials were difficult to analyze by the needle tracking algorithm and many of the results had to be discarded. Therefore a different lighting method has been acquired; LED backlight has been used to acquire a good contrast between the needle and the test environment, Fig 5.1.1(B). This has shown great improvement in needle tip tracking.

The control of the needle insertion procedure is currently done in a decentralized manner like shown in fig 5.1.2. The Elmo controllers are preprogrammed before the insertion with a certain motion profile. The parameters that are needed to run a motion profile are set in Matlab and send to the controllers once an insertion needs to begin. Previously all available motion profiles were set in the Elmo controllers by van Veen [8]. This Elmo code has been adapted and is now available to run any motion profile that is requested by the user from Matlab. The amount of rotations still depends on the parameters that are sent from Matlab previous to the insertion but multiple motion profiles can now be requested directly from Matlab. The rotational motion of the EMax22 is determined by an array of time values. These time values can be inserted into the Matlab code running the insertion manually or they can be extracted from simulations using the Simulink model.

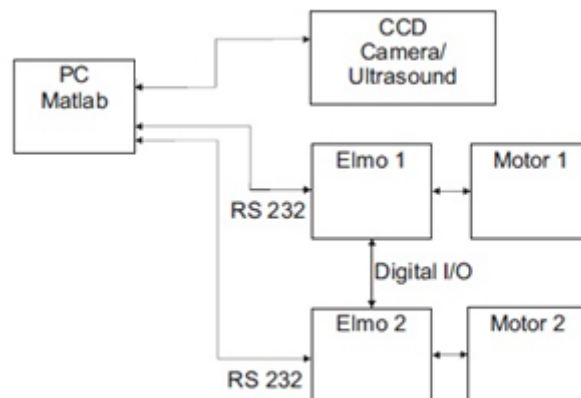


Fig 5.1.2 Control structure of the experimental setup [8].

The control system requires real-time visual feedback of the needle tip position and orientation to be able to steer it towards a target. However it has proven to be difficult to design a tracking algorithm to detect the needle tip in real-time. Because of the current control structure using motion profiles and the absence of real-time visual feedback, the current control system can only be used to plan a needle trajectory according to simulations. The extracted data from these simulations is then used as a feed forward by the needle insertion device.

The experiments, shown in section 5.2, have been performed using a 0.8 mm diameter Nitinol needle with a bevel-angle of 30°, see table 5.1 According to Y.Veen(2011) this type of needle resulted in the most deflection compared to thicker needles and needles with different tip shapes. No thinner needles or stiffer gelatins were used since this caused the needles to buckle during insertion. To prevent the current 0.8 mm needles from buckling, a guidance block and plug were used. They support the needle just before the point of insertion into the phantom. All insertions were done in a tissue phantoms consisting of 14.9% (mass percentage) gelatin and water. These samples were all made 24 hours before insertion took place. After mixing the substances (water and gelatin powder) at 64°C the samples were cooled over night and taken out of the refrigerator 4 hours before performing experiments. The 14.9% phantom has a elasticity of 35.5 KPa [8]. This value lies within the range of elasticity values that are found in prostate tissue in the human body [13].

Parameter	Symbol	Value
Elasticity	GPa	75
Diameter	D	0.8 mm
Bevel angle	A	30°
Length	L	170 mm

Table 5.2 Needle properties: (Simulation and experiments)

5.2 Experimental results

Needle insertion experiments were performed to validate and adapt the model/control algorithm and for acquiring data that would serve as an input to the model and control algorithm, section 5.3.1. In section 5.3.2 the deflection profile of the model is compared with experimental data. Since no real-time feedback is available yet the control algorithm could not be verified by real-time steering of the needle insertion setup. However, some feed-forward experiments, using both the model and control algorithm, have been performed as proof of concept, section 5.3.3.

5.2.1 Analyzing insertion data

The raw deflection data of three different needle insertion experiments are shown in fig 5.2.1. It shows the needle tip coordinates as measured by the tracking algorithm during single, double and triple bend insertion. Every needle insertion was performed by inserting the needle to an insertion depth of 130mm. The initial bevel-face orientation was downwards so all needle trajectories showed deflection along the positive y-axis.

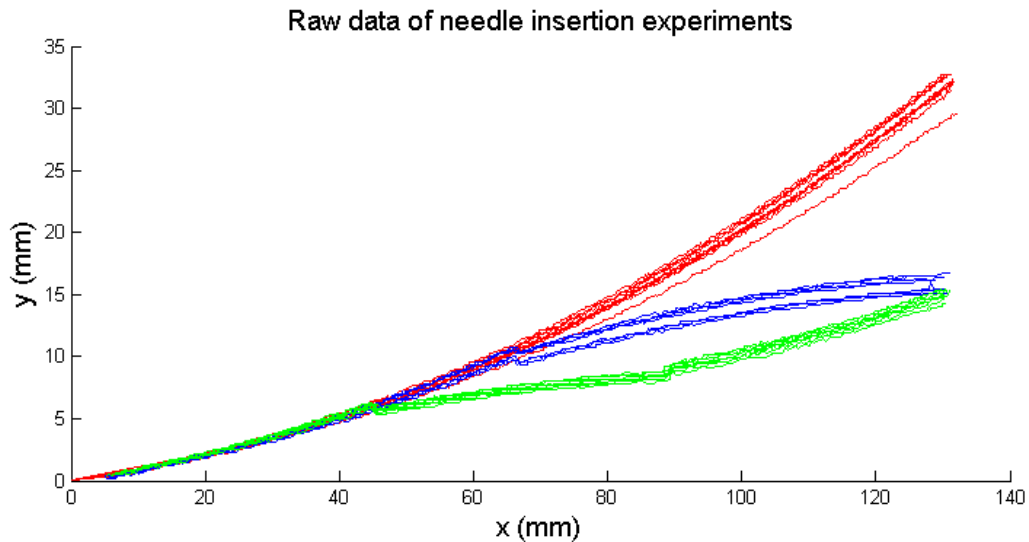


Fig 5.2.1 Raw deflection data of single(red), double(blue) and triple(green) bend needle insertion.

From these results, data has been obtained that was used as an input to the model. Also the model has been adapted to fit the experimental results. First of all, in the deflection model and steering algorithm the most important input is the radius of curvature. As mentioned in section 3, two different algorithms, shown in Appendix C, were used to determine the radii for the three types of needle insertion, Fig 5.2.1. This data has been analyzed to see if there is a trend to be noticed in the curvature data. Algorithm A fits a circular arc to the position data of the needle tip during the insertion. From such a circular arc, a constant radius of curvature can be determined. Algorithm B determines the curvature along a third (or higher order) polynomial function that is fit to the position data. The output of the algorithm is a variable radius along the insertion depth. The resulting constant and variable radii values are shown in table 5.2 and fig 5.2.2 respectively.

	Single bend Radii (mm)	Double bend Radii (mm)		Triple bend Radii (mm)		
Data range:	0-130 mm	0-65 mm	65-130 mm	0-43 mm	43-86 mm	86-130 mm
Insertion:	R_{sb}	R_{Db1}	R_{Db2}	R_{Tb1}	R_{Tb2}	R_{Tb3}
#1	408	392	414	396	425	355
#2	417	419	450	349	444	388
#3	458	437	412	393	444	366
#4	403	407	448	376	416	363
#5	410	461	526	336	384	358
#6	443	455	496	-	-	-
#7	419	466	502	-	-	-
#8	395	-	-	-	-	-
#9	416	-	-	-	-	-
#10	420	-	-	-	-	-
#11	444	-	-	-	-	-
Mean	416	434	464	370	423	366
σ	19	29	45	27	25	13

Table 5.2 Radii values for single and double bend insertion

As is shown in Fig 5.2.1 all needles showed similar curvature during the first 43 mm of the insertion. However, the radii from single bend (R_{sb}), double bend (R_{db}) and triple bend (R_{tb}) that have this deflection data in common differ from each other, see table 5.2. This difference is caused by the circular arc fitting algorithm. Plotting the circular arc fit over the insertion data showed that using Algorithm A is more reliable when fitting the arc to a bigger set of data points. For the triple bend curves, which consist of fewer data points, Algorithm A wasn't able to make a correct fit for every insertion. When fitting an arc to a smaller subset of data points, in many cases, the bending radius was estimated too small, within the range of 200-300 mm and plotting the resulting arc on the data points didn't show a good match. This flaw in the current fitting Algorithm hasn't been fixed in the time scope of this assignment and therefore some measurements had to be left out. These results should however match the data that is already shown in table 5.2 since from fig. 5.2.1 it can be concluded that all bends show similar curvature. Because of this inaccuracy in arc fitting, the data that is acquired from larger data sets (single bend curve, R_s) is assumed to be most reliable and these values will be used when fitting the model to experimental data.

For every set of insertion trials, a few insertions show a peak in their radius of curvature. This difference is most probably caused by out of plane deflection and therefore the needle will deflect less along the y-axis, which is observed by the camera. The radii of curvature don't show a trend that was initially expected (that the radius of curvature would increase with insertion depth). I assumed this to be true since the further the needle would get inserted into a tissue, the more force the tissue could exert on the needle that would prevent it from bending. This however does not seem to be the case.

A trend that does seem to be apparent is that the radius of curvature increases when the needle is deflecting downwards; in opposite direction of the initial bending curve. Observations of the initial needle shape showed that they are not perfectly straight so the change in curvature might be caused by this inaccuracy. However, it is clear that the curvature of the needle changes after a rotation has taken place.

Algorithm B didn't prove to be a good method for obtaining the radius of curvature during needle insertion. When changing the amount of data to which a polynomial function was fitted, the polynomial fit coefficients differed. Therefore also the curvature of the polynomial function differed along that insertion depth when compared with the curvature of a polynomial function which was acquired over a larger dataset. The curvature data over the full insertion depth of single bend needle insertion seems most reliable since it shows good comparison with the radii of curvature obtained by using Algorithm A, Fig 5.2.2. The variable radius acquired from both curves of double bend insertion doesn't show good comparison with the data from Algorithm A, Fig 5.2.3. The Algorithm showed similar results for triple bend insertion.

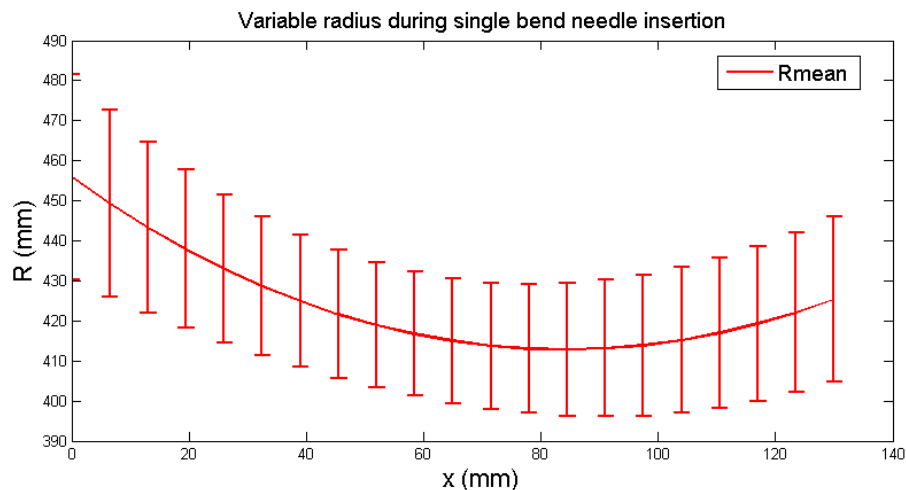


Fig 5.2.2 Mean variable radius of curvature and standard deviation during single bend insertion.

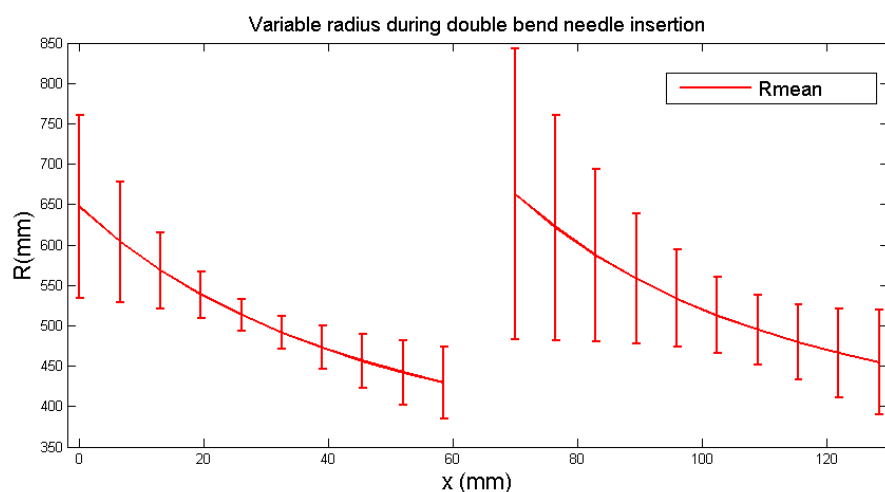


Fig 5.2.3 Mean variable radius of curvature and standard deviation during double bend insertion.

It can be concluded that, the radius of curvature changes when multiple bends/ rotations occur but that this change in curvature only shows slight difference in deflection. Since the radius of curvature over larger datasets (single bend) showed a better match with the experimental data, these values were used as an input to the model. ($R_{\text{mean}} = 416 \text{ mm}$)

What is also important is that the radius of curvature of the needle should be related to the tissue and needle parameters (e.g. elasticity, needle diameter, bevel-angle). The radius of curvature for similar insertions (e.g. single bend) in different tissue batches showed a difference in curvature as well. Therefore it is important to determine the radius of curvature before inserting a needle into a tissue simulant by knowing the needle and tissue parameters.

It has also been noticed that almost all insertions showed some initial angle, Θ_0 , which varied between $0.02 - 0.1 \text{ rad}$. This initial angle should be taken into account when comparing simulation results with experimental data and therefore it has been added as an input to the model. The most probable reasons for this initial angle are

1. The fact that the needle isn't perfectly straight and it has therefore the tendency to already curve into a direction; a preferred bending direction.
2. The needle tip tracking algorithm defines a visual detection box and the sides of this box don't always match with the sides of the gelatin sample. Therefore an initial angle (between the box and the sample) is measured according to the tracking algorithm.

5.2.2 Deflection data comparison

The experimental deflection data doesn't show perfect circular deflection and therefore differs from my model assumption. A comparison between the initial model ($R = 416$ mm) and the experimental data is shown in Fig 5.2.4 for double bend. The deflection profiles show resemblance during the first curve, after the rotation takes place the curves move apart which results in a final deflection error of 3.2 mm between the model ($y = 19.5$ mm) and the mean experimental deflection ($y = 16.3$ mm). This error is caused by irregularities that occur at needle rotation.

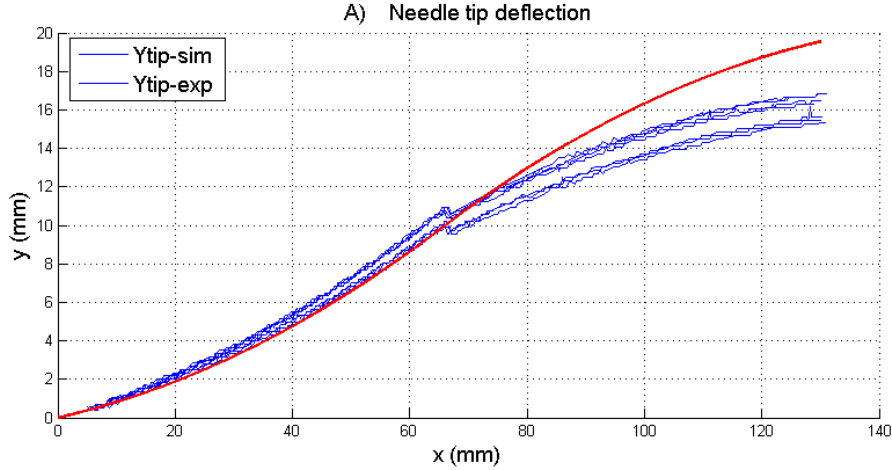


Fig 5.2.4 Double bend deflection. Model deflection (red) vs Experimental deflection (blue)

Fig 5.2.5 shows a zoomed image of the deflection data at rotation. The tracking algorithm tries to detect the precise needle tip and therefore the position of the needle tip shifts, Δd (0.5-0.8 mm), at rotation. This effect is shown in Fig 5.2.6. The model and control algorithm take the needle tip as a whole or simply show the deflection of the needle tip at its center of rotation so therefore this shift does not occur. By adapting the needle tracking algorithm this error could be corrected.

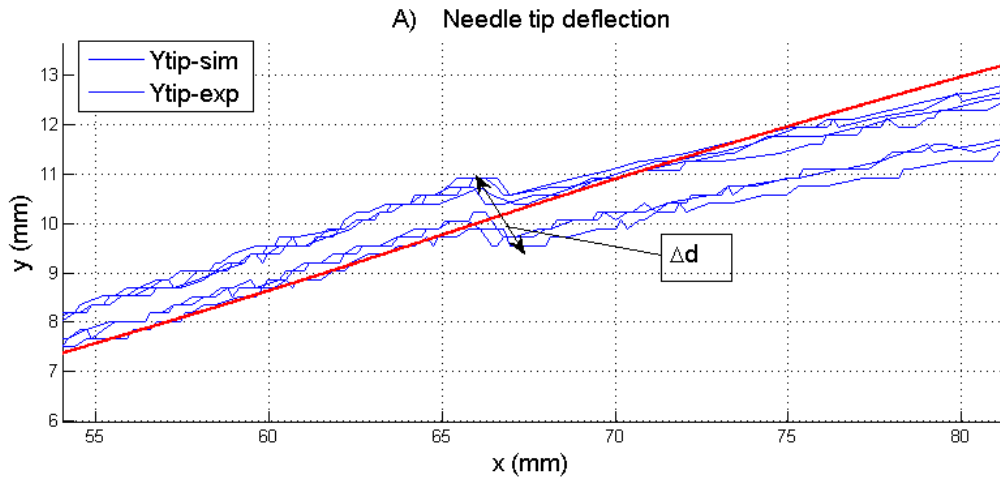


Fig 5.2.5 Needle tip deflection before and after rotation.

The biggest error between the model and reality however, arises because the actual needle deflection trajectory after rotation is not tangent to the one before rotation. Circular arcs that are tangent are assumed in the model and therefore there is an error in the departure angle after rotation, see Fig 5.2.7. It can be concluded that steeper bending curves can be made by the actual

needles because the trajectories before and after rotation of the needle are not tangent; a shift in the orientation angle, $\Delta\theta$, occurs at rotation. This difference in slope before and after rotation is most probably caused by the bending forces on the needle shaft that change when a rotation takes place. One more reason for the difference between the model and the experimental deflection is that the assumption that there is no tissue deformation during the insertion is incorrect. During insertion, the needle shaft will push the tissue sideways and this will affect the needle tip deflection, especially if the needle is rotated when it is already bend inside the tissue.

The model should be adapted to take these irregularities into account and to make a better approximation of the needle tip deflection. The control algorithm concept (steering by using R_C) is however still valid as will be shown in section 5.4 which will elaborate on adapting the model and improved needle steering.

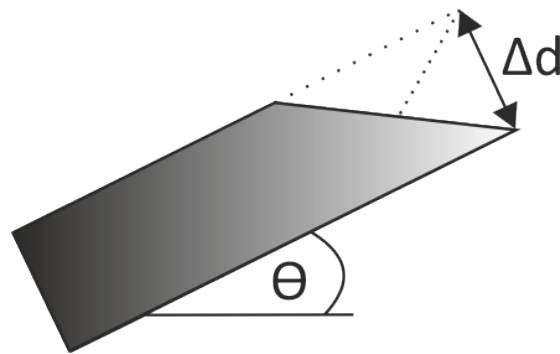


Fig. 5.2.6 Δd at needle rotation

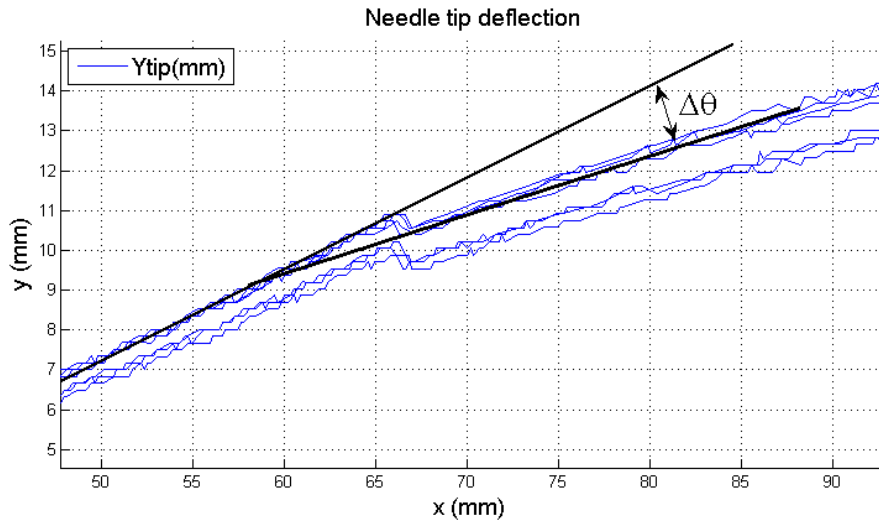


Fig 5.7 Difference in needle slope before and after rotation caused by $\Delta\theta$.

5.3 Proof of concept: Steering towards a target or along a path

No real-time visual feedback has been available up till now. To verify the concept of the control algorithm however, some feed-forward experiments have been performed. In these experiments both the simplified needle deflection model and the control algorithm are used to steer the needle setup. Error between the model and the actual deflection is to be expected, see section 5.2.2, but these experiments are performed to show the principle of steering along a path or steering towards a target. The experiments were performed by running simulations, extracting the rotational data and using this data as a feed forward for the setup. R is set at 416 mm and $R_c = 500$ mm. Steering of the needle towards a target was done for two different deflection profiles:

- Steering towards a single target
- Steering along a straight path defined by sub-targets

Steering towards a target

The target is located at (130,0) and the control algorithm determines the needle trajectory based upon the data coming from the deflection model. The resulting data is shown in Fig 5.3.1 where the solid black line is the estimated deflection data of the model and the red lines show the measured experimental deflection for 3 insertions. The experimental deflection shows a large difference in slope before and after rotations take place.

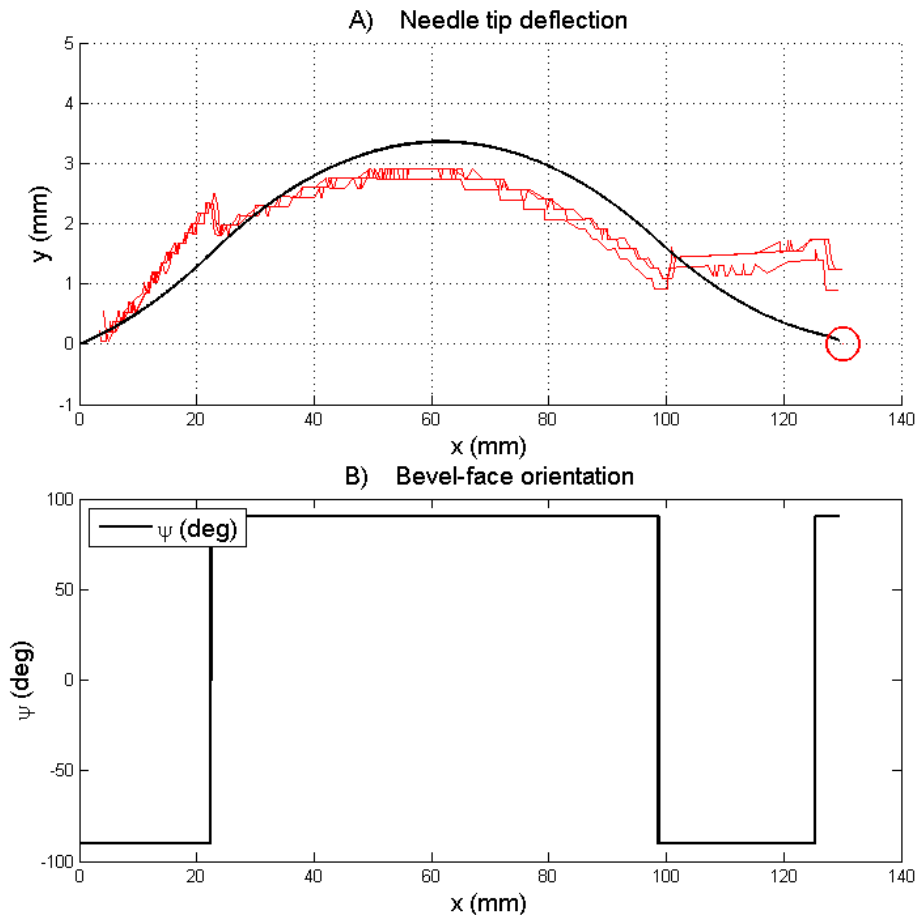


Fig. 5.3.1 Feed forward needle steering towards a target. Model deflection (black), Target location (round marker). Deflection data from three insertion trials (red).

The needle trajectories don't match because the model assumes tangential bending curves and because Θ_0 has to be perfectly matched to give a good comparison between the model and the actual needle deflection. The final deflection error at the target is 0.9 – 1.25 mm.

Steering along a linear path

The control algorithm steers the needle along a linear path towards (130,3). The error that arise at rotation are slightly averaged out because multiple rotations take place when following this linear path. However, the final deflection error at the last target is 0.2 – 1.2 mm. The resulting data is shown in Fig 5.3.2 where the solid black line is the estimated deflection data of the model and the red lines show the measured experimental deflection for 2 insertions.

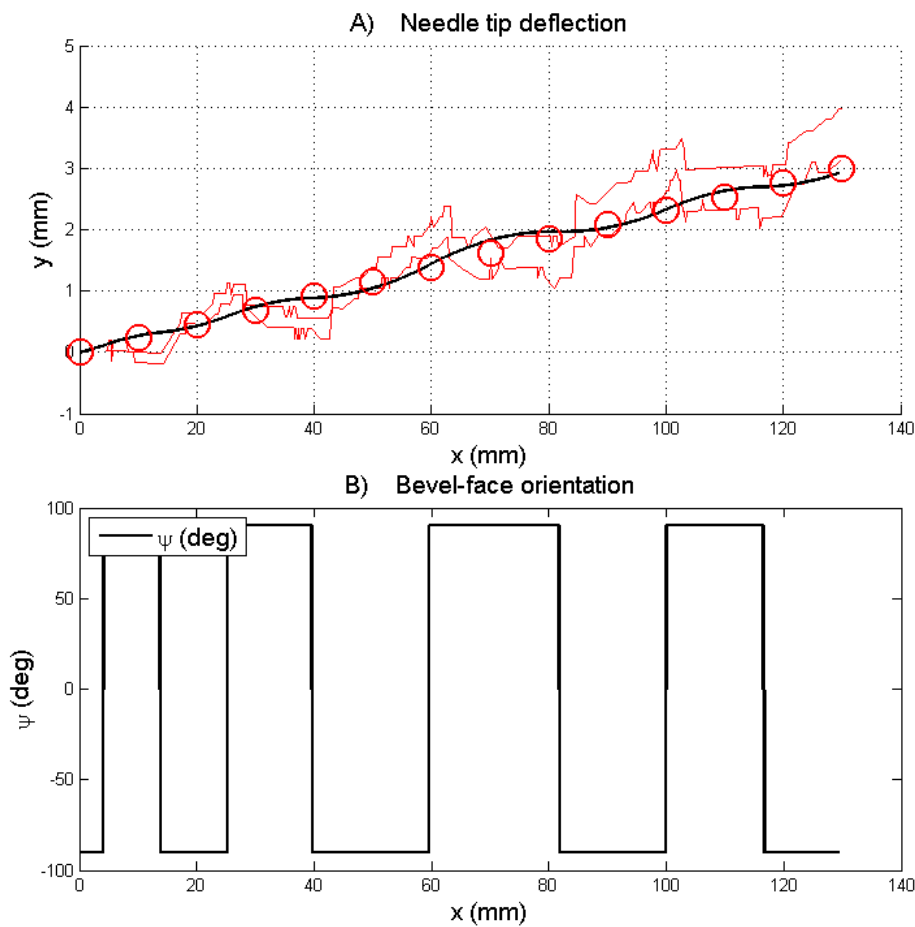


Fig. 5.3.2 Feed forward needle steering along a linear path. Model deflection (black), sub-target location (round markers). Deflection data from two insertion trials (red).

5.4 Model modification

The results from section 5.2/5.3 show that the initial model didn't match the experimental deflection and therefore it had to be modified. Experimental results, Fig 5.2.4 and Fig 5.2.5, have shown that the needle bending curves are not tangential to each other and because of a shift in angle $\Delta\theta$, the needle can make sharper turns. The needle deflection model has been adapted to take this into account by using a kinematic bicycle model, see Fig 5.4.1. The bicycle model is able to trace out arcs that don't have to be tangential to each other, which was shown by Webster et al.[4]. In this section it is shown that this model proves to be a better fit to the experimental data. The model and parameters are shown in Fig 5.4.2 [14].

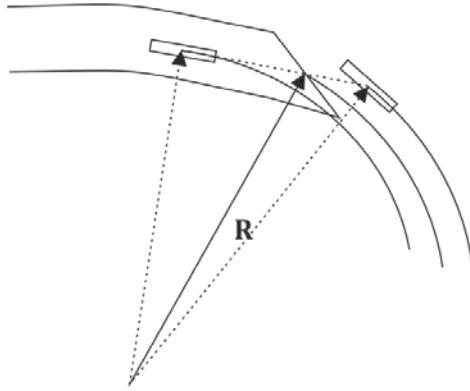


Fig. 5.4.1 Bicycle model for needle deflection.

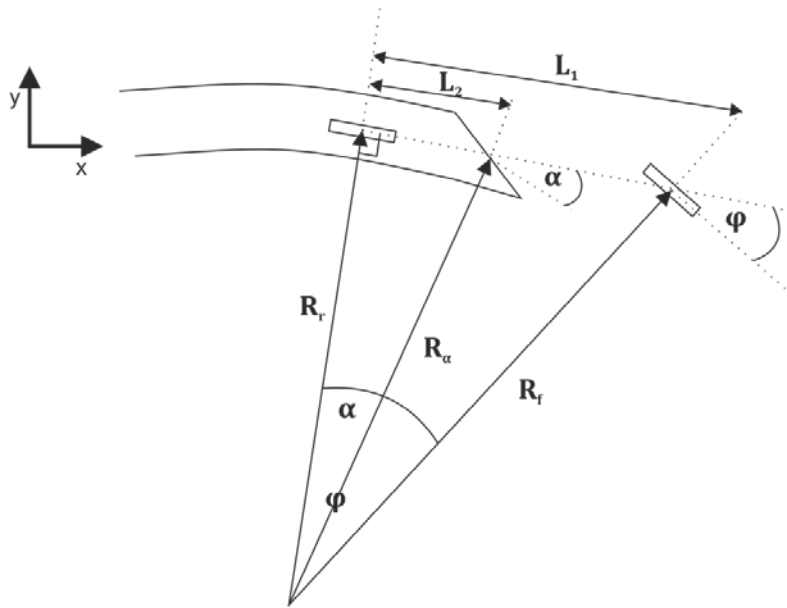


Fig 5.4.2 Bicycle model and parameters

R_r (rear wheel) $\approx R_f$ (front wheel) $\approx R_\alpha$ (needle tip) $\approx R$ and $L_1 \ll R$ & $L_2 \ll R$.

The radius of curvature of the needle (bicycle) is related to φ (the steering angle) and L_1 (length of the bicycle) according to

(6.1)

$$R = \frac{L_1}{\tan(\varphi)}$$

L_1 and φ are fitted to the radius of curvature that is obtained from the experimental data for single bend. L_2 is fitted to the experimental data to take into account $\Delta\theta$ and therefore approximate the needle deflection after rotation. The arcs that are traced out by the model are not tangential because of α which can be determined by

(6.2)

$$\tan(\alpha) = \frac{L_2}{L_1} * \tan(\varphi)$$

The steering angle is either positive or negative and therefore $\Delta\theta$ at rotation is equal to 2α (since the angle 'flips' from $-\alpha$ to $+\alpha$). The shift in angle is added to the equation of motion

(6.3)

$$\begin{bmatrix} V_x \\ V_y \\ \omega \end{bmatrix} = \begin{bmatrix} \cos(\theta + \tan^{-1}(\frac{L_2}{L_1} \cdot \tan(\varphi))) \\ \sin(\theta + \tan^{-1}(\frac{L_2}{L_1} \cdot \tan(\varphi))) \\ \frac{1}{R} \end{bmatrix} * V_{in}$$

and the resulting radius of curvature at L_2 is determined by

(6.4)

$$R_\alpha = \frac{L_2}{\sin(\alpha)} = \sqrt{L_2^2 + \left(\frac{L_1}{\tan(\varphi)}\right)^2} \approx R$$

The radii of curvature (R_r R_f R_α) only differ by 0.2% ± 0.05 because L_1 and L_2 (0-30 mm) are small compared to these radii (350-500 mm). Therefore these radii are considered equal.

The parameters in table 5.3 where obtained by empirically fitting the model to the deflection data of single, double and triple bend needle insertion. More research should be done on the relation between these parameters and the needle parameters (e.g. bevel-angle). In Fig 5.4.3 and Fig. 5.4.4 plots are shown of experimental needle tip deflection and needle tip deflection according to the model using the parameters from table 5.3.

Parameter	Value
R	416 mm
ϕ	0.06 rad
L_1	25 mm
L_2	12 mm

Table 5.3. Bicycle model parameter values.

The model shows a better match with the experimental data and therefore gives us a better approximation of needle deflection in the actual setup. The bicycle model stays overall within a standard deviation length from the mean experimental deflection. The second curve in double bend shows a bigger error because the mean experimental deflection shows the needle tip location which has shifted at rotation. The concept that is used by the control algorithm is also still valid for the bicycle model which is shown in Fig. 5.4.5.

More rotations of the needle would occur when it gets closer towards the target because the model assumes to make steeper bending curves at every rotation. This has for now been solved by adding extra hysteresis, up to 0.3 mm, to the control system which is similar to adding a time delay. However, because steeper bending curves can be made, $\Delta\theta$ should also be taken into account by the algorithm. $\Delta\theta$ might also differ per rotation that takes place and therefore more research should be done on why and how this change occurs. For now, the current control algorithm has been implemented and according to simulations this seemed to show good result in needle steering, see Fig 5.4.5.

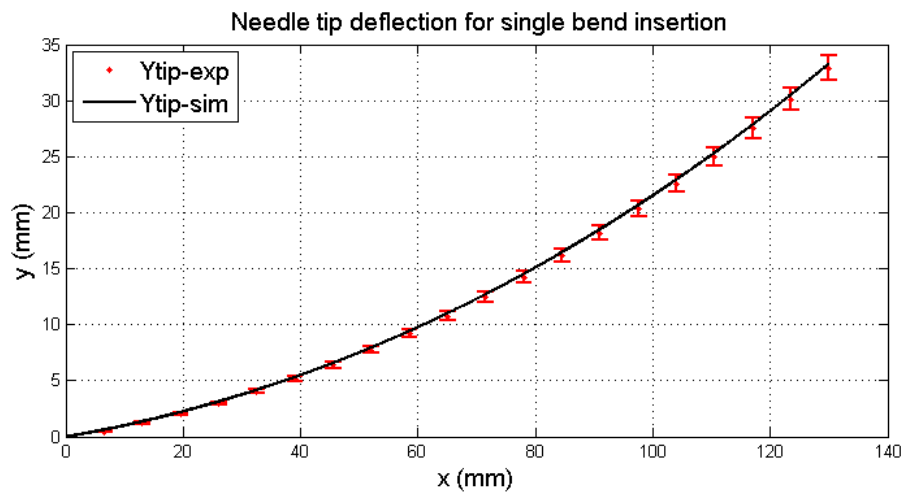


Fig 5.4.3 Needle tip deflection for single bend insertion of experimental data and bicycle model.

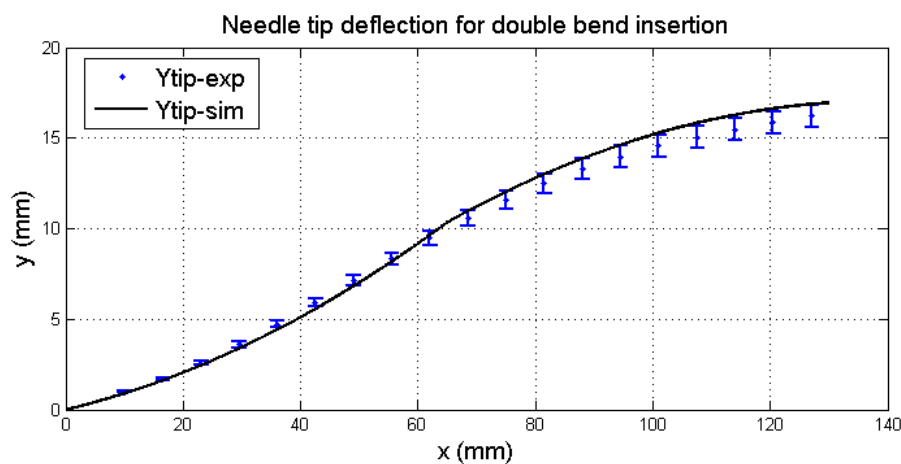


Fig 5.4.4 Needle tip deflection for double bend insertion of experimental data and bicycle model.

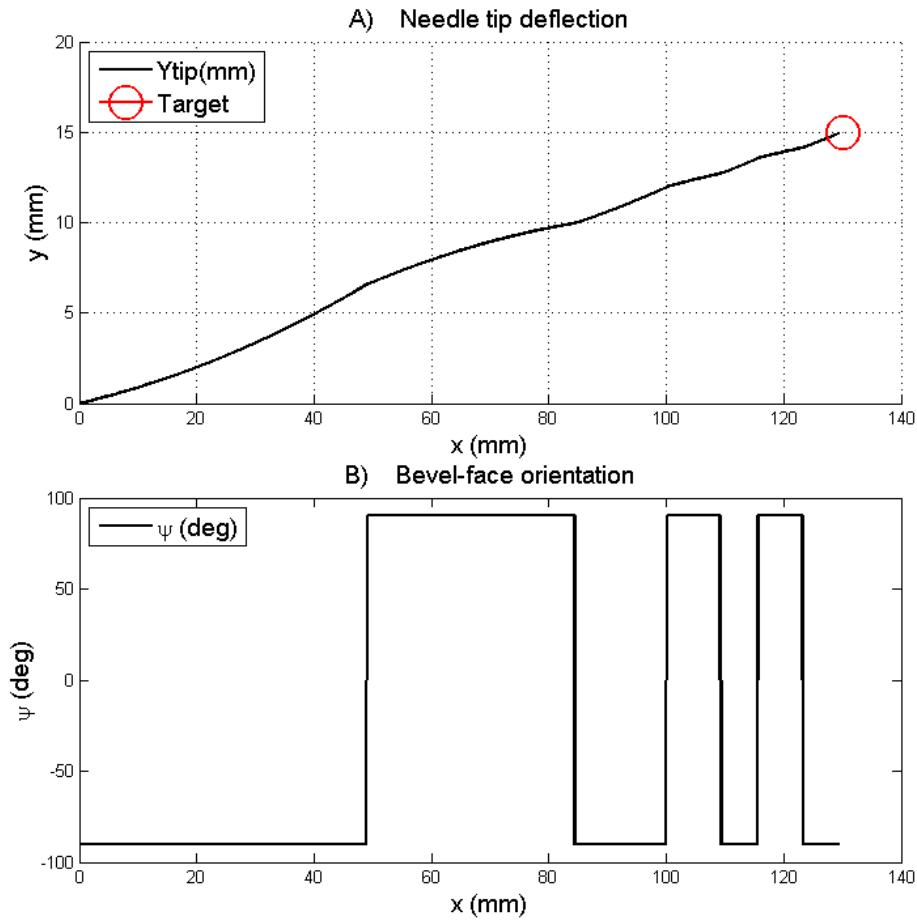


Fig. 5.4.5 Needle steering using the control algorithm on the bicycle model.

5.5 Conclusion on results and validation

The radii of curvature have been obtained from single, double and triple bend insertion. Analyzing the insertion data has shown that the radius of curvature changes when rotation takes place but that the radius obtained from single bend insertion matched best with experimental data. The experimental needle deflection didn't match the estimated deflection from the initial model which assumed needle deflection along tangent circular arcs. The error occurred because the actual needle curves are not tangent to each other and because the experimental results show a shift in the needle tip position at rotation of the needle. This shift is caused by the fact that the tracking algorithm tracks the precise needle tip. Although the initial model doesn't match the needle deflection after rotation takes place, feed-forward experiments have been performed to show the concept of needle steering with the algorithm in MR insertion setup. The initial deflection model has later on been modified to fit the experimental deflection of double and triple bend insertion. Simulations using the modified 'bicycle' model have shown that needle steering with the control algorithm is still valid.

6. Conclusion and Recommendations

Based on the work done and described in the chapters of this report, some conclusions have been drawn on the project. Furthermore some recommendations are put forward; these could be used for further research in the needle steering project.

6.1 Conclusion

The goal of this assignment, to design a real-time feedback control system for steering a flexible needle during insertion into a soft tissue phantom has mainly been fulfilled.

6.1.1 Modeling

A 2D model has been derived to approximate needle deflection in a tissue phantom. This model was initially based upon previous research which showed that a flexible needle will bend with a constant radius of curvature. Experimental data from the insertion setup of the Medical Robotics group has been used to obtain the radii values from different insertion trials. These radii values were obtained using two different methods, either fitting a circular arc or determining the change in radius of curvature during insertion. The results from these algorithms have shown that the deflection model can use a constant radius of curvature to give an approximation of needle deflection.

6.1.2 Model validation

The model has been validated using single, double and triple bend insertion. The initial deflection model did not match the experimental data after rotation had taken place. Therefore this model has been modified to include the changes that occurred in needle bending after rotation. The modified 'bicycle' model does match the experimental data and stays within a millimeter from the mean experimental deflection.

6.1.3 Controller design

A 2D Needle control algorithm has been designed that can steer the needle towards specific targets and it can steer the needle along a predefined path of sub-targets. The control system can also follow predefined path like a straight line or a polynomial. To make sure that uncertainties in needle bending (radius of curvature) don't affect the steering capability of the control system steering decisions were based upon a 'funnel shape' which is defined by a radius that is larger than observed in the experimental data. This radius takes into account all fluctuations in the curvature that might happen during insertion. The main note is that this radius should always be bigger than the actual radius might become. Using this method the needle will be rotated multiple times to keep the needle tip in range to reach the target. From simulations it can be concluded that a constant radius of curvature could be used to steer needles showing variation in curvature. Needle steering with a constant radius can be applicable also in our situation if the radius of curvature is chosen large enough to make the effect of change in radius negligible. However, steering the needle could be optimized when the control system takes into account this variation.

6.1.4 Verification

Verification of the controller is yet to be done. This could not be acquired since no real-time visual feedback was available which should be used by the algorithm to steer the needle. However, using feed forward, proof of concept has been done. These experiments were performed using the model that already showed a difference in needle tip deflection with the experimental results. The error between the assumed deflection and the experimental data went up to 1.5 mm.

6.1.5 Research & Experimental work

Good documentation saves a lot of trouble and is the key to good research.

6.2 Recommendations

Since the Medical Robotics group will continue its research on flexible needle insertion, I would like to make some recommendations based on my experience in this assignment.

6.2.1 Needle insertion guidance

Needle insertion guidance should be implemented to prevent needle buckling and to make sure that the needle is inserted perfectly straight into the phantom. As shown in section 5, the initial angle of orientation Θ_0 has a lot of influence on the needle insertion trajectory. This initial angle is also caused by the needle tracking 'box', a predefined area in the video frames in which the algorithm searches for the needle tip. The sides of this box should be aligned with the gelatin containers of the experimental setup.

6.2.2 Real-time needle tracking

Real-time needle tracking should be acquired. This real-time visual feedback is needed for the controller to work and for the algorithm to be verified. To prevent a non-linear shift in tip deflection at needle rotation, the tracking algorithm should track the centre of rotation at the needle tip or the needle tip as a whole.

6.2.3 2D to 3D

The 2D model and control should be upgraded to 3D. During insertion experiments the needles always moved out of plane. This influenced my measurements of the deflection of the needle tip and therefore also the radii values may have varied. To take this movement into account, needle control should be used in 3D which can prevent the needle from moving out of plane or actually steering towards a target that is located out of the 2D plane. The concept of the control algorithm is still valid in 3D but then a 3D funnel will be defined originating from the needle tip.

6.2.4 Bevel face direction

Currently the initial direction of the bevel-face has to be set manually which adds error to the needle deflection in the (x,y) plane. Out of plane movement is mainly caused by inaccurate positioning of the bevel face. The direction of the bevel face should be clearly marked and the initial direction should be measured using sensors.

6.2.5 Centralizing needle insertion control

Centralizing needle insertion control via Matlab by using XPC target. XPC target allows for the elmo controllers to receive real-time input and this is required for real-time needle steering. Real-time needle steering could also be acquired outside of Matlab by using C++.

6.2.6 Path planning

The deflection model and control algorithm could give an approximation of the needle trajectory towards a predefined target. Obstacle avoidance could currently be done by defining multiple sub-targets that define a path. To include obstacle avoidance however, a path planning algorithm should be designed. At every position during the insertion of a needle all possible paths towards the target should be determined from which one will be chosen. This depends on the decision criteria.

6.2.7 Graphical User Interface (GUI)

Design of a GUI in which you can select the target on screen and a path planning algorithm will make sure that an optimal needle path is chosen towards this target avoiding obstacles that are also shown on screen.

6.2.8 Variable radii of curvature

Change in radius of curvature should be taken into account during needle insertion. The radius of curvature should be updated during insertion and the control algorithm should take this change into account when defining the collision 'funnel' on which it bases steering decisions. More experiments should be done on acquiring the radii of curvature for different samples and these radii of curvature should be related to some tissue properties such as elasticity. This information is needed for needle steering through different tissue samples. Experiments with multi layer gelatins could be performed to verify how well the algorithm works with parameter (radii) variation during insertion.

6.3 Acknowledgement

I would like to thank Momen Abayazid, Roy Roesthuis, Jasper Keuning and Rob Reilink for their support and discussions about the assignment. Special thanks for Japser's coffee machine which was always around to supply us with a fresh cup of coffee to keep us going.

References

1. D. Glozman, M Shoham, Image-Guided Robotic Flexible Needle Steering, IEEE Transactions on robotics, vol. 23 no. 3, June 2007.
2. K. Hauser, R. Alterovitz, N. Chentanez, A. Okamura, K. Goldberg, Feedback Control for Steering Needles Through 3D Deformable Tissue Using Helical Paths, Robotics: Science and Systems, June 2009.
3. S. Misra, K.B.Reed, B.W. Schafer, K.T. Ramesh and A.M. Okamura, Mechanics of Flexible Needles Robotically Steered through Soft Tissue, The International Journal for Robotics Research, 2010.
4. R.J. Webster III, J. Seob Kim, N. J. Cowan, G. S. Chirikjian, A. M. Okamura, Nonholonomic Modeling of Needle Steering, Department of Mechanical Engineering, The Johns Hopkins University, The International Journal of Robotics Research, Vol 25, No. 5-6, May-June 2006.
5. R. Alterovitz, A Lim, K Goldberg, G. S. Chirikjian, A.M. Okamura, Steering Flexible Needles Under Markov Motion Uncertainty, International Conference on Intelligent Robots and Systems (IROS), Aug 2005, pp.120-125.
6. N. Abolhassani, R. V. Patel, F. Ayazi, Minimization of needle deflection in robot-assisted percutaneous therapy. The international journal of medical Robotics and computer assisted surgery, 2007, 3, 140-148.
7. R. J. Roesthuis, Y.R.J. van Veen, A. Jahya and S. Misra, Mechanics of Needle-Tissue Interaction, MIRA- Institute of Biomedical Technology and Technical Medicine, University of Twente, The Netherlands.
8. Y.R.J. van Veen, MSc. Thesis on 'Needle and Soft-tissue simulant interactions: Experimental design and observations', Faculty of Engineering Technology and Laboratory of Biomechanical Engineering, University of Twente, The Netherlands, 2011.
9. <http://www.cookmedical.com/uro/dataSheet.do?id=1860> 18-06-2011
10. <http://www.mathkb.com/Uwe/Forum.aspx/matlab/86580/curvature-and-radius-of-curvature-of-a-plane-curve> 08-07-2011
11. C. S. Stewart, a. Bradley, c. Leibovich, a. L. Weavera and m. M. Lieber, prostate cancer diagnosis using a saturation needle biopsy technique after previous negative sextant biopsies, department of urology and section of biostatistics, mayo clinic, rochester, minnesotaaaccepted 9 february 2001.
12. N. Abolhassani, r. Patel, and m. Moallem. Control of soft tissue deformation during robotic needle insertion. Minimally invasive therapy and allied technologies, 15(3):165{176, 2006.
13. T. A. Krouskop, t. M. Wheeler, f. Kallel, b. S. Garra, and t. Hall. Elastic moduli of breast and prostate tissues under compression. Ultrasonic imaging, 20(4):260{274, 1998.
14. <http://sites.google.com/site/bikephysics/english-version/2-geometry-and-kinematics> 30-08-2011

Appendices

A Matlab & Simulink

A1 *mfile with initial parameters*

```
clear all

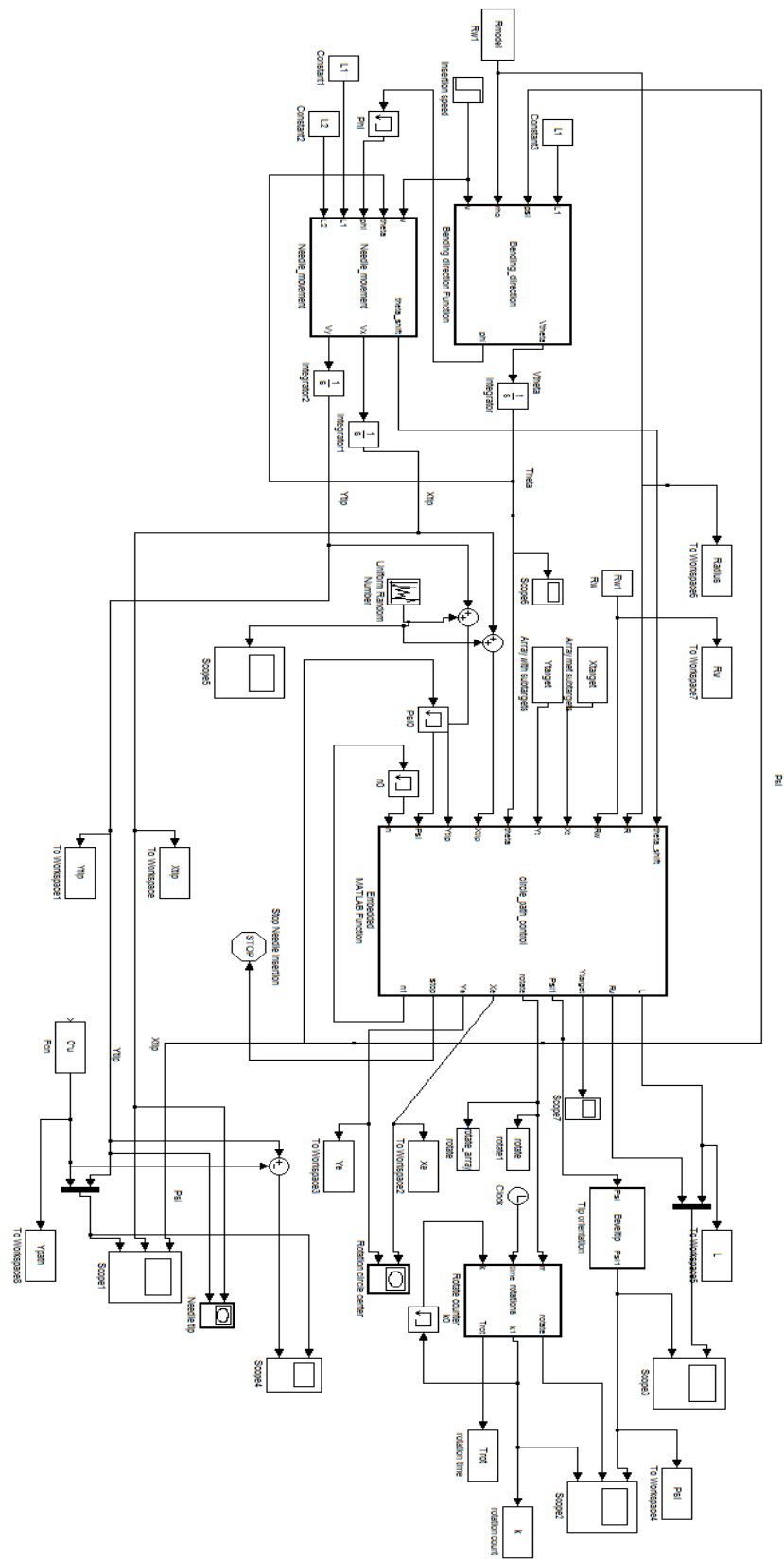
%% Set parameters
load('Rmean_single_bend_constant.mat')

% Initial position & orientation
X0 =0;
Y0 =0;
theta0 = 0.05;
% Max time of the simulation
tmax = 30;
% Timestep
time_step = 0.01;
% Insertion velocity in mm/s
v = 5;
% Timeline of simulation
t = (0:0.01:tmax)';
% X-vector
X = v*t;
% X- and Y-coordinates of the (sub)target(s).
% Xtarget = [20 40 60 80 100 130];
Xtarget = [130];
% Xtarget = [0:10:130];
% Ytarget = 1/13.*Xtarget;
% Ytarget = -4/4425.*Xtarget.^2-130*(-4/4225).*Xtarget;
% Ytarget = 1/1600.*Xtarget.^2-1/100.*Xtarget;
% Ytarget = 0.*Xtarget;
% Ytarget = [2 3.5 3 2 1 0];
Ytarget = [15];
% Ytarget = -3/4225.*Xt.^2+390/4225.*Xt;
% First target
n0 = 1;
% Initial value rotation counter
k0 = 0;
% Initial value of the needle tip orientation: false = beveltip directed
% downwards and true = beveltip directed upwards.
Psi0 = false;
% Needle bending radius (R), defined in main_deflection
Rmodel = 440;
% Largest needle bending radius
Rw1 = 500;

Rh = 0.5;

L1 = 25;
L2 = 12;
phi = atan(L1/Rmodel);
```

A2 Simulink model



A3 Control algorithm

```

function [L,Rv,Ytarget,Psi1,rotate,Xe,Ye,stop,n1] =
circle_path_control(R,Re,Xt,Yt,theta,Xtip,Ytip,Psi,n)

% Hysteresis (radius of the hysteresis circle around the target)
Rh = 0.2;

% Determine next (sub)target
Xtarget = Xt(n);
Ytarget = Yt(n);

% Distance of needle tip to target
Ds = sqrt((Xtarget-Xtip)^2 + (Ytarget-Ytip)^2);

% Target has been (nearly) reached, continue towards next target or stop
% insertion.
if(Ds <= 0.5 || Xtip>Xtarget)
    if(n == length(Xt))
        stop = true;
        n1 = n;
    else
        stop = false;
        n1 = n+1;
    end
else
    stop = false;
    n1 = n;
end

% Actions to be taken when the beveltip is facing downwards
if(Psi == false)
    %Centre of error margin circle
    Xe = Xtip + Re*sin(theta);
    Ye = Ytip - Re*cos(theta);
    %Distance of the error margin circle to the final target
    L = sqrt((Xtarget-Xe)^2 + (Ytarget-Ye)^2);
    %Rotation point, about which the needle is currently rotating
    Xc = Xtip - R*sin(theta);
    Yc = Ytip + R*cos(theta);
    Rv = Re-Rh;
    if(L<=Rv && Ds > 0.5)
        Psi1 = true;           %rotate
        rotate = true;
    else
        Psi1 = false;         %do nothing
        rotate = false;
    end
% Actions to be taken when the beveltip is facing upwards
else
    %Centre of error margin circle
    Xe = Xtip - Re*sin(theta);
    Ye = Ytip + Re*cos(theta);
    %Distance of the error margin circle to the final target
    L = sqrt((Xtarget-Xe)^2 + (Ytarget-Ye)^2);
    %Rotation point, about which the needle is currently rotating
    Xc = Xtip + R*sin(theta);
    Yc = Ytip - R*cos(theta);
    Rv = Re-Rh;
    if(L<=Rv && Ds > 0.5)
        Psi1 = false;         %rotate
        rotate = true;
    else
        Psi1 = true;          %do nothing
        rotate = false;
    end
end
end

```

B Elmo code

The part of the Elmo code that was used for the multiple rotations motion profile, used by the Emax22 controller, is shown below with extra comments. The version that is uploaded into the Elmo controllers doesn't include comments since the controller has a maximum capacity for the amount of characters that can be stored in its memory.

```
int Trot[30]           // Rotation time array
int rotations          // Amount of rotations

##MR
MO=0;                  // Motor is off
PX=0;                  // Define current position as reference point.
MO=1;                  // Motor is on
SP=10240;              // Define rotational speed counts/sec
PR = 1024;             // Add 1024 counts to PX. 2048 counts equals full rotation
// Begin needle insertion
until (IB[18]);
OB[1]=0;
RR=3;
for i=1:rotations      // Iteration, amount of rotations
wait(Trot[i])          // Wait a specified time in Trot until next rotation
BG                     // Begin rotation
if(PR == 1024)         // Rotate up to the next PX value
until(PX>=1024)
else
until(PX<=0)
end
ST                     // Stop motion
PR = -PR;              // Next rotation will be opposite to the current one
End
// Back to initial position
wait(Holdtime)
if(PR == 1024)
else
BG
until (PX<=0)
ST
end
MO=0;                  // Turn of motor
OB[1]=1;               // Reset output
exit
```

C Algorithms used to determine the needle curvature.

Algorithm A: Fitting a circular arc to experimental data. [10]

```
mx = mean(xdata); my = mean(ydata);
X = xdata - mx; Y = ydata - my; % Get differences from means
dx2 = mean(X.^2); dy2 = mean(Y.^2); % Get variances
t = [X,Y]\(X.^2-dx2+Y.^2-dy2)/2; % Solve least mean squares problem
a0 = t(1); b0 = t(2); % t is the 2 x 1 solution array [a0;b0]
r = sqrt(dx2+dy2+a0^2+b0^2); % Calculate the radius
a = a0 + mx; b = b0 + my; % Locate the circle's center
curv = 1/r; % Get the curvature
y = -sqrt(r^2-(xdata-a).^2)+b;
```

Reflection on Algorithm A

Analyzing the insertion data using Algorithm A shows different results when fitting a circular arc to a small subset of data points compared to fitting a circular arc to the full data set. Plotting these fits with the experimental data showed that the fit over the full data set was a better match with the data. This problem arises because the algorithm tries to make a fit only to the data set which is selected and it does this by also moving the circle's center. To compare the radius of subsections with the one from the full data set, the circle's center should be kept in place. This problem hasn't been solved in the time span of this assignment and therefore the radius of curvature that was determined from the total insertion data is considered to be true since it shows a better fit with the data.

Algorithm B: Fitting a polynomial function to the experimental data and determine the curvature along this polynomial.

```
% The curvature of some function y = f(x) is defined as:
%          dy^2/d^2x
% k = -----
%      ( 1 + (dy/dx)^2 )^3/2
%
% The radius of the circle (R) is the defined as:
% R = 1/k
%
% I have a function describing the deflection of the needle
% (yExp), using the formula above I can calculate the radius of
% curvature for needle bending.
```

Polynomial function

```
yExp(i,:) = subs(vExpPol(i,:), x, xExp);
% Calculate the first and second derivative of this function
dvExp1 = diff(vExpPol(i,:), x, 1);
ddvExp1 = diff(vExpPol(i,:), x, 2);
% With these derivatives the curvature of the function can be determined
k1 = abs(ddvExp1) / ( 1 + dvExp1^2 )^(3/2);
% From the curvature, the radius of curvature is determined.
R_sym = 1/k1;
```

Reflection on Algorithm B

The algorithm didn't prove to be a good method for obtaining the radius of curvature during needle insertion. When changing the amount of data to which a polynomial function was fitted, the polynomial fit coefficients differed. Therefore also the curvature of the polynomial function differed along that insertion depth when compared with the curvature of a polynomial function which was acquired over a larger dataset. The difference in radius of curvature between polynomial fits over data belonging to the same dataset went up to 50% as is shown in figure C1 for single bend insertion. There was also the problem that some polynomial fits consisted of linear parts. The radius of curvature for these sections spiked up to infinity and this made measurements very unclear. Therefore it proved difficult to determine an accurate mean radius of curvature for the multiple bend insertion data. Algorithm B hasn't further been used as an input for the needle deflection model because of these reasons.

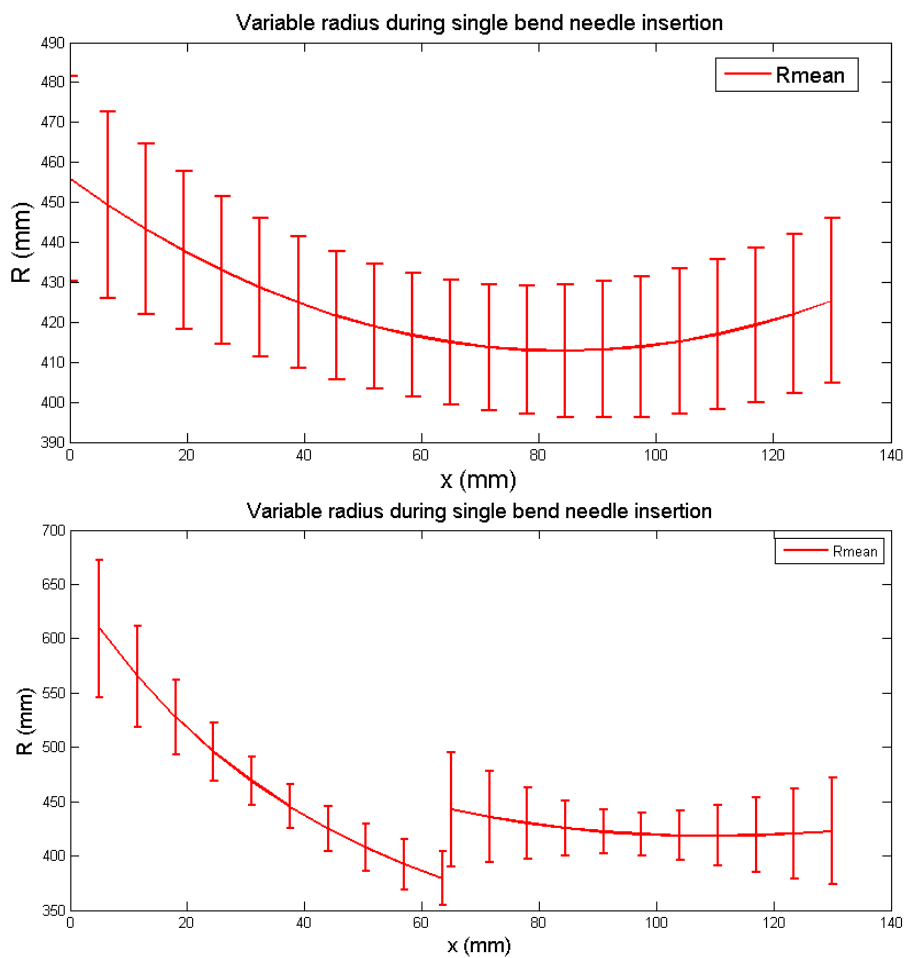


Fig C1 Mean variable radius of curvature during single bend insertion. Top figure shows the radius for a single polynomial fit. The bottom figure shows the radius for two separate fits.

The Root mean square error (RMSE) for single bend insertions and both curved sections of double insertions was determined at

$$\text{RMSE} = 0.08331 \pm 0.02.$$

D Preliminary control algorithm

To make sure that the needle would stay between two boundaries the assumed shape of the movement of the needle (circles of constant radius) had to be taken into account by the controller. For the needle to stay exactly between the boundaries, Θ has got to be equal to zero when reaching the allowed deflection. Therefore the control system had to take into account the current angle Θ and the angular velocity ω of the next circular path that the needle will follow once it rotates,

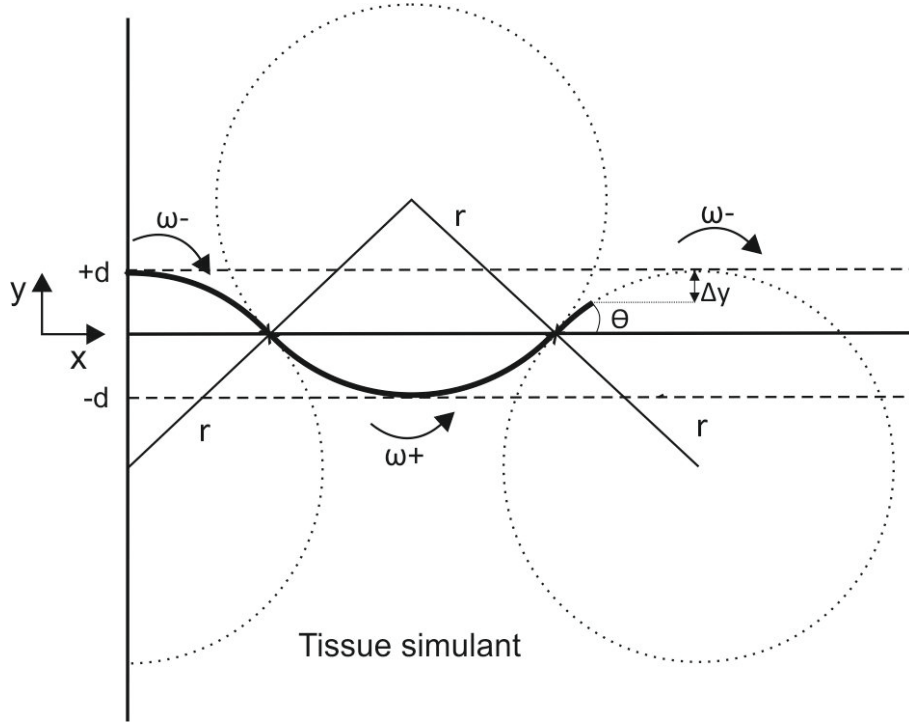


Fig D1. Needle control by using distance to the next boundary, the angle between the inertial frame and the needle tip and the amount of angular velocity. The crosses resemble the rotation points.

The control algorithm is based on equation (1-4). The controller checks when it is able to decrease Θ to exactly zero at the next boundary by using the angular velocity of the next circular movement. This ω is equal to the negative of the current angular velocity. The difference in time (Δt) over which this has to take place is determined by the mean value of the velocity in the translational direction ($\overline{V_y}$) and the amount of deflection that it will take for the needle tip to reach the next boundary (Δy), eq (4).

The rotation point of the needle can be determined by

$$(1) \quad \theta - \omega \cdot \Delta t = 0$$

With

$$(2) \quad \Delta t = \frac{\Delta y}{\overline{V_y}}$$

Where the mean value of V_y in this case is equal to

$$(3) \quad \overline{V_y} = \frac{1}{2} \cdot V \cdot \sin |\theta|$$

Combining the equations will give us the following condition for rotating the needle

$$(4) \quad \theta - \omega \cdot \frac{\Delta y}{\frac{1}{2}V \cdot \sin |\theta|} = 0$$

If the needle tip is located between the boundaries it will make sure that θ is equal to zero once it reaches the boundary it was moving towards. It does this by rotating in the opposite direction once the conditions stated in (7) is met.

However, I would like to be able to steer the needle in every direction in the 2D(x,y) plane. In order to make the needle follow any planned path in this plane, the boundary conditions needed to be adjusted for every specific path. The angle between the planned path and the inertial frame may differ during the insertion so this has to be taken into account to determine the new boundaries. With the use of this angle, Alfa (α), we can rotate the IF to a path fixed frame (PFF), which coordinates will follow the motion of the chosen path. The controller will then relate the needle movement to this PFF, shown in fig. D2 for a linear path in plane.

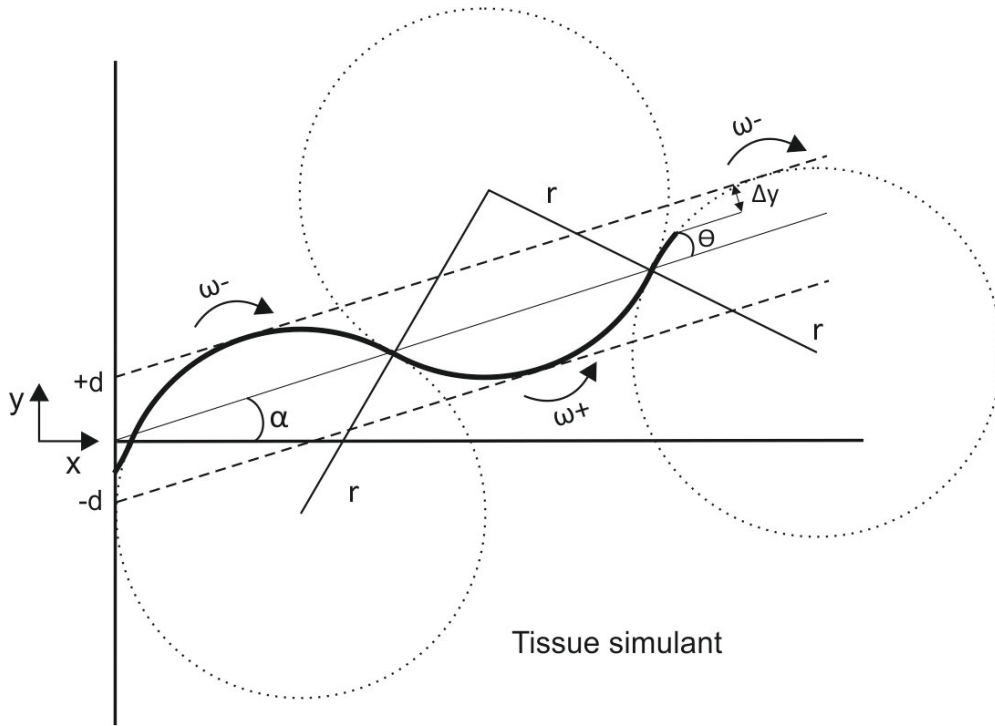


Fig. D2 Path of the needle tip when following a linear path in plane.

From the slope of the planned path, the value for α is determined.

$$(8) \quad \alpha = \tan^{-1} \frac{\Delta y}{\Delta x} = \tan^{-1} \frac{\delta y}{\delta x}$$

The angle between the needle tip and the PFF is equal to

$$(9) \quad \theta' = \theta - \alpha$$

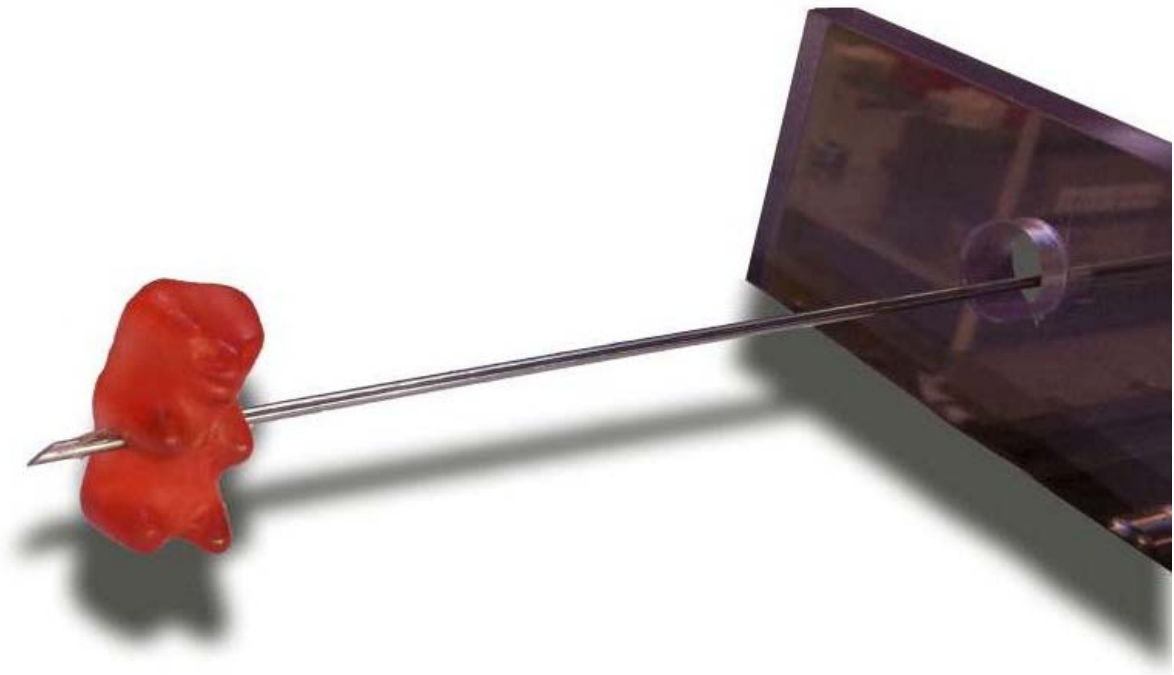
All other relations mentioned before will remain the same but the values of the parameters will now depend on α .

For linear paths there is no change in slope, so the controller was able to keep the needle tip precisely between the stated boundary deflections (d).

If I kept the boundaries quite close to the planned path, $d = 0.5$ mm, the needle was also able to follow the polynomial paths quite well.

However these paths don't take the needle properties into account. The ideal path planner would use the circular shaped movement of the needle to determine which path should be taken. In this way the needle wouldn't have to be rotated as much. This would result in less stress between the needle and the patient's tissue and it is therefore more patient friendly.

Moreover, this control system will just follow the directed path that has been aimed at the target, and rotate as much as possible to stay in between the defined boundaries so you can't control the amount of rotations that are made.



UNIVERSITY OF TWENTE.