A Bevel-Tip Needle Path Tracking: A Model Predictive Approach

Lambert Ren

***Abstract*—This paper focuses on a real-world issue that arises in medical scenarios. Specifically, we explore the application of model predictive control (MPC) to the problem of tracking the path of a bevel-tip needle model. The model can be linearized as two totally different system when considering different curvatures. To demonstrate the stability of these linear MPC controllers, a stability analysis has been performed on both of the models. Performance of a time-varying system with various parameters have been compared to find a optimal solution for the MPC controller. Finally, a 3D simulation created by Python is created to further prove the controller successfully achieve the desired outcome.**

I. Introduction

Percutaneous punctures are minimally invasive surgical procedures that are commonly used for diagnosing and treating lesions located in hard-to-reach areas. They are frequently employed for a variety of medical purposes, such as biopsy, injection, radiofrequency ablation, drug delivery, and more [1]. Despite the benefits, there are several factors that can cause the needle tip to deviate from its intended target. These factors may include tissue inhomogeneity, tissue deformation, as well as the surgeon's hand-eye incoordination and fatigue [2]. To enhance the controllability of puncture needles, a new approach involving the use of a bevel-tip flexible needle has been proposed [3]. The accuracy is improved via precise planning and controlling due to the flexibility and controllable of the bevel-tip needle. The common approach of controlling the bevel-tip needle is the PID controller [4]. However, the use of this type of controller may result in limitations to the accuracy of the surgical procedure.

This paper introduces a novel approach, a linear model predictive control (MPC) system, to control the bevel-tip needle during surgical procedures. This system is designed to enable the needle to accurately follow predetermined paths.

A. Bevel-tip needle dynamics

As stated by Huo [5], the bevel-tip needle can be characterized as a nonlinear dynamic system, with the following equations governing the motion:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{x}_k + \dot{\mathbf{x}} \cdot \triangle dt \\ \dot{\mathbf{x}} &= \mathbf{A}(\mathbf{x}) \begin{bmatrix} v \\ \omega \end{bmatrix} \\ \mathbf{x} &= \begin{bmatrix} p_x & p_y & p_z & \alpha & \beta & \gamma \end{bmatrix}^T, \end{aligned} \tag{1}$$

Lambert Ren is master students at TU Delft, The Netherlands. E-mail addresses: z.ren-4@student.tudelft.nl.

where $\mathbf{x}_m$ represents the state of needle at time step k, which is composed of the coordinates p_x, p_y, p_z in 3D space, as well as the attitude angles α, β, γ representing the angle rotated separately with respect to the x, y, z axis shown in Fig. 1. The needle's motion is determined by an insertion speed v, as well as a rotational speed ω, which corresponds to the rotation around the z-axis, these are also plotted

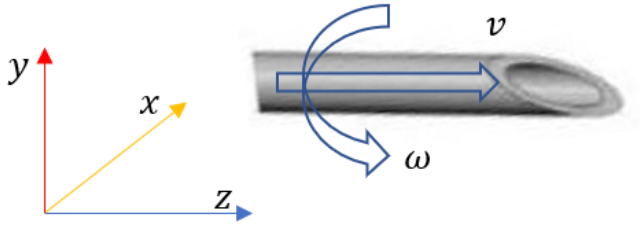

Fig. 1. Dynamics model of the bevel-tip needle.

Moreover, in the case of the bevel-tip needle model, there is typically a parameter, curvature κ, that also determines the path followed by the needle tip, as shown in Fig. 2. With the help of curvature and rotation speed, the tip can rotate further flexibly. As a part of the shape of the needle, curvature is generally set to a specific number initially [6].

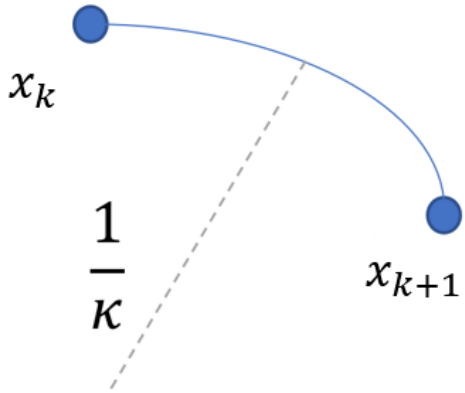

Fig. 2. The curvature of the arc followed by bevel-tip needle.

Assuming that the needle tip is moving in its own body coordinate system, its velocity can be represented as

$$\mathbf{V}^b = v\mathbf{V_1} + \omega\mathbf{V_2}, \tag{2}$$

where $\mathbf{V}^b$ is the velocity of the needle tip in its own body

coordinate system, and $\mathbf{V}_1$, $\mathbf{V}_2$ can be described as the following,

$$\begin{aligned}\mathbf{V}_1 &= \begin{bmatrix}0 & 0 & 1 & \kappa & 0 & 0\end{bmatrix}^T \\ \mathbf{V}_2 &= \begin{bmatrix}0 & 0 & 0 & 0 & 0 & 1\end{bmatrix}^T\end{aligned} \tag{3}$$

The translation between the velocity of the needle tip in the world coordinate system and itself body coordinate system can be described as

$$\mathbf{V}^b = \mathbf{J}\mathbf{V}^w, \tag{4}$$

where $\mathbf{V}^w$ is the velocity of needle tip in the world coordinate system, $\mathbf{J}$ is the translation matrix $\mathbf{J} = \begin{bmatrix}\mathbf{R}^T & 0_{3\times3} \\ 0_{3\times3} & \mathbf{R}_\Omega\end{bmatrix}$. In this matrix, $\mathbf{R}^T$ is the rotation matrix between the world coordinate system and needle body coordinate system, $\mathbf{R}_\Omega$ is the coefficient matrix of Euler kinematics equations of rigid body motion [7].

Therefore, integrating all the equations given previously, we can finally reach to the kinematics model between two time steps:

$$\dot{\mathbf{x}} = \mathbf{A}(\mathbf{x})\begin{bmatrix}v \\ \omega\end{bmatrix} = \begin{bmatrix} sin\beta & 0 \\ -cos\beta sin\alpha & 0 \\ cos\alpha cos\beta & 0 \\ \kappa cos\gamma sec\beta & 0 \\ \kappa sin\gamma & 0 \\ -\kappa cos\gamma tan\beta & 1\end{bmatrix}\begin{bmatrix}v \\ \omega\end{bmatrix} \tag{5}$$

B. Linearization and discretization of needle tip system

To linearize and discretize the dynamic model, we need to derive the equation based on two assumptions. First, we assume that the needle is stiffness in axial direction. Therefore, the needle tip will rotate exactly the same as the needle tail, where the power of rotation generates. We also assume that the needle is a rigid body model. In this case, the needle body moves exactly according to the trajectory traveled by the needle tip. Under these two assumptions, we can only take the needle tip as the robot model in our MPC system and only track the position and orientation of the needle tip during the experiment.

Given the significant impact of the curvature value (κ) on the dynamics model, this study will consider both scenarios: when κ is set to 0 and when it is not. When κ is set to 0, the bevel-tip needle model can be regarded as a standard needle, similar to those commonly used for vaccination injections. This is because the two attitude angles, α and β, will remain 0 as long as the needle is strictly inserted vertically into the skin. Therefore the displacement in both x and y directions will also stay unchanged during the whole process. However, when κ is set to a positive value, for example $\frac{1}{150}mm^{-1}$, the path of the needle will take the form of a spiral curve. Both two conditions will be considered in this study.

1) $kappa = 0$: As mentioned earlier, when setting $\kappa = 0$, the system can be simplified to a standard needle. If we remove all the other parameters stay unchanged while keeping the remaining dynamic model, it can be described by

$$\begin{aligned}\dot{p}_z &= v \\ \dot{\gamma} &= \omega.\end{aligned} \tag{6}$$

In this case, the tip model becomes a linear model with only two states p_z and γ. Therefore, the discrete needle model can be derived as

$$\begin{aligned}p_z(k+1) &= p_z(k) + v \times dt \\ \gamma(k+1) &= \gamma(k) + \omega \times dt,\end{aligned} \tag{7}$$

where p_z is the needle tip's position in the world coordinate system relative to the beginning point, γ is the needle tip's total rotation angle in around z-axis in the world coordinate system accumulated from the initial point, v is the axial velocity, ω is the rotational velocity with respect to the z-axis and dt is the time step between two states in time series. This formalizes a time-invariant needle tip system.

2) $\kappa \neq 0$: When κ is set to a positive value, the bevel-tip needle model is a highly-nonlinear model. To derive a linear model from the nonlinear dynamics, linearization is performed around the trajectory using the Taylor's expansion. Supposing that for state $\mathbf{x}_k$, the closet point on reference path has a state $\mathbf{x}_r$ with a reference control $\mathbf{u}_r$. Therefore, the system can be formalized as a new error state $\epsilon = \begin{bmatrix}\epsilon_x & \epsilon_y & \epsilon_z & \epsilon_\alpha & \epsilon_\beta & \epsilon_\gamma\end{bmatrix}^T$, which describes the error between the robot state $\mathbf{x}_k$ and the reference state $\mathbf{x}_r$. Finally, the dynamic system can be discretized as

$$p_x(k+1) = v_r cos(\beta_r)\epsilon_\alpha + sin(\beta_r)\epsilon_v \times dt + p_{xr}, \tag{8}$$

$$\begin{aligned}p_y(k+1) =& -v_r cos(\alpha_r)cos(\beta_r)\epsilon_\alpha \\ &+ v_r sin(\alpha_r)sin(\beta_r)\epsilon_\beta \\ &- cos(\beta_r)sin(\alpha_r)\epsilon_v \times dt + p_{yr},\end{aligned} \tag{9}$$

$$\begin{aligned}p_z(k+1) =& -v_r sin(\alpha_r)cos(\beta_r)\epsilon_\alpha \\ &- v_r cos(\alpha_r)sin(\beta_r)\epsilon_\beta \\ &+ cos(\beta_r)cos(\alpha_r)\epsilon_v \times dt + p_{zr},\end{aligned} \tag{10}$$

$$\begin{aligned}\alpha(k+1) =& \kappa v_r cos(\gamma_r)sec(\beta_r)tan(\beta_r)\epsilon_\beta \\ &- \kappa v_r sin(\gamma_r)sec(\beta_r)\epsilon_\gamma \\ &+ \kappa cos(\gamma_r)sec(\beta_r)\epsilon_v \times dt + \alpha_r,\end{aligned} \tag{11}$$

$$\begin{aligned}\beta(k+1) =& \kappa v_r cos(\gamma_r)\epsilon_\gamma \\ &+ \kappa sin(\gamma_r)\epsilon_v \times dt + \beta_r,\end{aligned} \tag{12}$$

$$\begin{aligned}\gamma(k+1) =& -\kappa v_r cos(\gamma_r)sec^2(\beta_r)\epsilon_\beta \\ &+ \kappa v_r sin(\gamma_r)tan(\beta_r)\epsilon_\gamma \\ &- \kappa cos(\gamma_r)tan(\beta_r)\epsilon_v \times dt \\ &+ \epsilon_\omega \times dt + \gamma_r.\end{aligned} \tag{13}$$

where the state of the closet reference point is $\mathbf{x}_r = \begin{bmatrix} p_{xr} & p_{yr} & p_{zr} & \alpha_r & \beta_r & \gamma_r \end{bmatrix}^T$ and the reference control input at this point is $\mathbf{u}_r = \begin{bmatrix} v_r & \omega_r \end{bmatrix}^T$. This formalizes the model into a time-varying needle tip system.

C. System dynamics

After the linearization and discretization performed in the previous section in both situations, the system can be transformed into the linear system:

$$\mathbf{x}_{k+1} = f(\mathbf{x}, \mathbf{u}) = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}. \tag{14}$$

Since the state and sparse matrix differs from two situations, they will be separately introduced in the following part.

1) $\kappa = 0$: In this situation, only the displacement on z-direction and rotation angle around z-axis need to be taken into account for the system dynamics. Therefore, the state in each time step can be described by

$$\mathbf{x} = \begin{bmatrix} p_z & \gamma \end{bmatrix}^T, \tag{15}$$

and the input to the system can be described as

$$\mathbf{u} = \begin{bmatrix} v & \omega \end{bmatrix}^T. \tag{16}$$

The corresponding matrix $\mathbf{A}$ and $\mathbf{B}$ is given as,

$$\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{17}$$

$$\mathbf{B} = \begin{bmatrix} dt & 0 \\ 0 & dt \end{bmatrix} \tag{18}$$

2) $\kappa \neq 0$: Under this situation, all six states need to be taken into account as they are all time-varying. Therefore, the state here will be modified to

$$\mathbf{x} = \begin{bmatrix} p_x & p_y & p_z & \alpha & \beta & \gamma \end{bmatrix}^T. \tag{19}$$

The control input remains the same as the one used previously in Equation 16. And the sparse matrix used in this state space model can be derived from the linearized system in Section I.B:

$$\begin{aligned}
\mathbf{A}[0,3] &= v_r cos(\beta_r) \\
\mathbf{A}[1,3] &= -v_r cos(\alpha_r) cos(\beta_r) \\
\mathbf{A}[1,4] &= v_r sin(\alpha_r) sin(\beta_r) \\
\mathbf{A}[2,3] &= -v_r sin(\alpha_r) cos(\beta_r) \\
\mathbf{A}[2,4] &= -v_r cos(\alpha_r) sin(\beta_r) \\
\mathbf{A}[3,4] &= \kappa v_r cos(\gamma_r) sec(\beta_r) tan(\beta_r) \\
\mathbf{A}[3,5] &= -\kappa v_r sin(\gamma_r) sec(\beta_r) \\
\mathbf{A}[4,5] &= \kappa v_r cos(\gamma_r) \\
\mathbf{A}[5,4] &= -\kappa v_r cos(\gamma_r) sec^2(\beta_r) \\
\mathbf{A}[5,5] &= \kappa v_r sin(\gamma_r) tan(\beta_r)
\end{aligned} \tag{20}$$

$$\mathbf{B} = \begin{bmatrix} sin(\beta_r) \times dt & 0 \\ -cos(\beta_r) sin(\alpha_r) \times dt & 0 \\ cos(\beta_r) cos(\alpha_r) \times dt & 0 \\ \kappa cos(\gamma_r) sec(\beta_r) \times dt & 0 \\ \kappa sin(\gamma_r) \times dt & 0 \\ -\kappa cos(\gamma_r) tan(\beta_r) \times dt & dt \end{bmatrix} \tag{21}$$

To be mentioned, all the other values not mentioned in Equation. 20 in matrix $\mathbf{A}$ are 0. Two different linear model predictive controllers will be built based on these two dynamics systems introduced here.

II. Model predictive control design

In this section, two model predictive controllers (MPC) designed to separately control the motion of the bevel-tip needle model in two different situations will be introduced. As introduced earlier, when $\kappa = 0$, the model used here is a linear time-invariant system, while the model will be a typical linear time-varying system when $\kappa \neq 0$. These two model will be separately introduced in the later parts.

It is worth noting that the dynamics systems are built based on the assumptions that the needle is stiff in axial direction and the it is also a rigid body model. Therefore, only the position, orientation and velocity of the needle tip will be tracked in this study.

1) $\kappa = 0$: For this linear time-invariant system, we select the control horizon as $N = 8$. The state constraints in this MPC controller can be divided into two parts. The first part is related to the system dynamics between two stages, which is illustrated in the Equation 14. Finally, the first stage for the needle model is set as the initial position. To be mentioned, there is a shifting in the coordinate system for this project, the origin is set to the starting point where the needle tip is at the beginning. Therefore, in general, the state constraints $\mathbf{x} \in \mathbb{X}$ are:

$$\begin{aligned}
\mathbf{x}_{k+1} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\
\mathbf{x}_0 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T
\end{aligned} \tag{22}$$

where $\mathbf{x}$ is the state with respect to the origin point described above.

The control input constraints in our MPC controller is mostly related to the limitation of the power generated for the acceleration of the needle tip and the torque for the angle rotated around z-axis. Therefore, there are upper bounds and lower bounds for both the linear and angular acceleration. Moreover, there will also be a limitation on the velocity of the needle tip, which is generally a rather low value in practical use as it will be rather painful for the patient if the needle is inserting with a rather high speed. In general, the control input constraints $\mathbf{u} \in \mathbb{U}$ are:

$$\begin{aligned} v &\leq v_{max} \\ \omega &\leq \omega_{max} \\ \omega &\geq \omega_{min} \\ v_{k+1} - v_k &\leq a_{max} \times dt \\ v_{k+1} - v_k &\geq a_{min} \times dt \\ \omega_{k+1} - \omega_k &\leq \alpha_{max} \times dt \\ \omega_{k+1} - \omega_k &\geq \alpha_{min} \times dt \end{aligned} \tag{23}$$

The cost function of this MPC model can be generally described as [8]:

$$\begin{aligned} J(\mathbf{x}_0, \mathbf{u}) = \sum_{k=0}^{N-1} \ell(\mathbf{x}(k), \mathbf{u}(k)) + V_f(\mathbf{x}(N)) \\ s.t. \mathbf{u} \in \mathbb{U}, \mathbf{x} \in \mathbb{X} \end{aligned} \tag{24}$$

To be mentioned, the sate here $\mathbf{x}$ refers to the error between needle tip's position and the destination point. The stage cost $\ell(x, u)$ and the terminal cost $V_f(x)$ are defined as:

$$\begin{aligned} \ell(\mathbf{x}, \mathbf{u}) &= \mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{u} \\ V_f(\mathbf{x}) &= \mathbf{x}^T \mathbf{P} \mathbf{x} \end{aligned} \tag{25}$$

where $\mathbf{Q} = diag(\begin{bmatrix} 10 & 10 \end{bmatrix})$, $\mathbf{R} = diag(\begin{bmatrix} 1 & 1 \end{bmatrix})$ and $\mathbf{P}$ is the solution of discrete algebric Riccati equation (DARE), which is solved numerically using function "scipy.linalg.solve_discrete_are". Since $\mathbf{Q}$ and $\mathbf{R}$ is time-invariant in this situation, the matrix $\mathbf{P}$ is also time-invariant here. Moreover, there will also be a terminal constraint guaranteeing the stability of the controller, which can be written as

$$\mathbf{x}(N) \in \mathbb{X}_f. \tag{26}$$

The terminal set is designed as a sub-level set of the terminal cost $\mathbb{X}_f = \{x \in \mathbb{R}^3 | \mathbf{x}^T \mathbf{P} \mathbf{x} \leq c\}$, where c is a hyperparameter, and this will further discussed in Section 3.

2) $\kappa \neq 0$: When curvature is set to a positive value, the model is a linear time-varying system as illustrated before. Hereby, the control horizon is set to $N = 8$ to guatantee a more precise tracking. Comparing with the state constraints when $\kappa = 0$, the system dynamics between two stages is a time-varying system and there is another limitation on the state β, since there is a trigonometric function appearing as the denominator in the equation between stages for α. To be mentioned, there is still a shifting in the coordinate system for this project, the origin is again set to the starting point where the needle tip is at the beginning. Therefore the new state constraints can be written as:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{A}(t)\epsilon + \mathbf{B}(t)\epsilon_u + \mathbf{x_r} \\ \beta &\neq (k + \frac{1}{2})\pi \\ \mathbf{x}_0 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \end{aligned} \tag{27}$$

where $\mathbf{x}$ is the state with respect to the origin point in this coordinate system, ϵ is the error state between the closet reference point on path and the needle tip's position, ϵ_u is the error control input between the reference control input on that point and the needle tip's input, $\mathbf{x_r}$ is the reference state of that point on which system is linearized.

The control input constraints in this time-varying MPC controller remain the same as those used in the previous time-invariant controller, which is described in Equation 23. In the meantime, the cost function used here is also same as the one used previously. The only difference is that the terminal cost which is shown in Equation 25, will also be a time varying function. This is because that the matrix $\mathbf{P}$ here is the solution to DARE, while matrix $\mathbf{A}$ and matrix $\mathbf{B}$ is time-varying in this situation. To be mentioned, the two weighting matrix here is set to $\mathbf{Q} = diag(\begin{bmatrix} 0.01 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \end{bmatrix})$, $\mathbf{R} = diag(\begin{bmatrix} 100 & 10 \end{bmatrix})$. Finally, terminal set is also time-varying, designed as a sub-level set of the terminal cost $\mathbb{X}_f = \{x \in \mathbb{R}^3 | \mathbf{x}^T \mathbf{P}(t) \mathbf{x} \leq c\}$.

Taking everything discussed before into account, the model predictive control problem is finally formalized as a time-varying system.

III. Asymptotic stability

In this section, asymptotic stability of both two scenarios will be discussed separately since one system is time-invariant and the other is time-varying. Different theorem in [8] will be used to prove the stability for these two different systems. Beforehand, design of the terminal set is first introduced.

A. Terminal set $\mathbb{X}_f$

The terminal set $\mathbb{X}_f$ is an invariant constraint admissible set for $\mathbf{A}_K \mathbf{x}$ where $\mathbf{A}_K = \mathbf{A} + \mathbf{B}\mathbf{K}$ and $\mathbf{K} = -(\mathbf{B}^T \mathbf{P} \mathbf{B} + \mathbf{R})^{-1} \mathbf{B}^T \mathbf{P} \mathbf{A}^T$ is the optimal control gain defined by infinite horizon LQR problem. As discussed in the previous section, The terminal set is designed as a sub-level set of the terminal cost $\mathbb{X}_f = \{\mathbf{x}^T \mathbf{P} \mathbf{x} \leq c\}$, where $\mathbf{P}$ is time-invariant when $\kappa = 0$ and $\mathbf{P}$ is time-varying when $\kappa \neq 0$. Therefore, this is a control invariant set.

1) $\kappa = 0$: The hyperparameter is selected as $c = 10$ when the system is time-invariant. It can be proved that this terminal set is a control invariant set with the following equations,

$$\begin{aligned} \mathbf{x}^{+T} \mathbf{P} \mathbf{x}^+ &= (\mathbf{A}_K \mathbf{x})^T \mathbf{P} \mathbf{A}_K \mathbf{x}, \\ \mathbf{P} &= \mathbf{A}_K^T \mathbf{P} \mathbf{A}_K + \mathbf{Q}_K. \end{aligned} \tag{28}$$

Combining these two equations,

$$\mathbf{x}^{+T} \mathbf{P} \mathbf{x}^+ = \mathbf{x}^T \mathbf{P} \mathbf{x} - \mathbf{x}^T \mathbf{Q}_K \mathbf{x} \leq c \tag{29}$$

Since there are only two states in this system, the terminal set is an ellipse in 2D plot, with two dimensions separately in pz and γ.

2) $\kappa \neq 0$: The hyperparameter is also selected as $c = 10$ when the system is time-invariant. And it can be again proved that this terminal set designed is a control invariant set with the same method used previously. As there are six states in this system, the terminal set here is a ellipsoid.

B. Proof of stability for system when $\kappa = 0$

As discussed before, the system is time-invariant under this situation, where both $\mathbf{A}$ and $\mathbf{B}$ are 2 by 2 matrix. Considering that the controllability matrix $\mathbf{W}_c = [BAB]$ has a rank of 2, this linear system is controllable. As described in the Section 2.4[8], the **Theorem 2.19** proves the MPC stability when there are terminal constraints within the controller, which needs to satisfy **Assumption 2.2**, **Assumption 2.3**, **Assumption 2.14** and **Assumption 2.17** in [8].

(i) **Assumption 2.2** (Continuity of system and cost)
Considering that the both the stage cost $\ell(\mathbf{x}, \mathbf{u})$ and the terminal cost $V_f(\mathbf{x})$ are continuous as they are all in quadratic form within the cost function. Moreover, when both state and control input are set to 0, all the function with quadratic form will have a value of 0, which means that the system satisfies $f(0,0) = 0$, $\ell(0,0) = 0$, $V_f(0) = 0$. This assumption is therefore satisfied.

(ii) **Assumption 2.3** (Properties of constraint sets)
As discussed in the last section about the state constraints and control input constraints, set $\mathbb{Z}$ is therefore closed. The terminal set $\mathbb{X}_f$ designed as a sub-level set is compact with a quadratic form. Moreover, these two sets both contain the origin point, which is the equilibrium point of the tracking. This assumption is therefore satisfied.

(iii) **Assumption 2.14** (Basic stability assumption)
There are two properties to prove within this assumption. To prove the assumption 2.14(a), $f(\mathbf{x}, \mathbf{u})$ is selected to be as $\mathbf{x}_K$, which is the solution to Discrete Algebraic Riccati Equation (DARE), $\mathbf{x}_K = \mathbf{A}_K\mathbf{x}$. This selection definitely satisfy the first acquirement $f(\mathbf{x}, \mathbf{u}) \in \mathbb{X}_f$. Combing the quadratic form of described previously in this report, the left part of the second inequality equation in the book can be rewritten as,

$$V_f(f(\mathbf{x}, \mathbf{u})) - V_f(\mathbf{x}) = (\mathbf{A}_K\mathbf{x})^T\mathbf{P}\mathbf{A}_K\mathbf{x} - \mathbf{x}^T\mathbf{P}\mathbf{x} \quad (30)$$

Since there is relationship between $\mathbf{A}_K$ and $\mathbf{P}$, $\mathbf{Q}_K$ and K described in DARE, the left part can be further derived as

$$\begin{aligned} V_f(f(\mathbf{x}, \mathbf{u})) - V_f(\mathbf{x}) &= \mathbf{x}^T(\mathbf{P} - \mathbf{Q}_K)\mathbf{x} - \mathbf{x}^T P\mathbf{x} \\ &= -\mathbf{x}^T\mathbf{Q}\mathbf{x} - (K\mathbf{x})^T\mathbf{R}K\mathbf{x} \end{aligned} \quad (31)$$

Also considering that $\mathbf{u} = K\mathbf{x}$. It is obvious that the final equation just describes the stage cost discussed previously. Therefore, the second requirement is also satisfied, $V_f(f(\mathbf{x}, \mathbf{u})) - V_f(\mathbf{x}) \leq -\ell(\mathbf{x}, \mathbf{u})$. And it has been proved previously that the designed terminal set $\mathbb{X}_f$ is a control invariant set.
To prove the assumption 2.14(b), we need to find two $\mathcal{K}_\infty$ functions, which is both strictly-increase and approximate infinite while x going infinite. For the function $\alpha_1(|x|)$, it can be set to $\mathbf{Q}\mathbf{x}^2$, since the part $\mathbf{u}^T\mathbf{R}\mathbf{u}$ is of quadratic form within the stage cost $\ell(\mathbf{x}, \mathbf{u})$. Therefore, the first requirement is satisfied. As to function $\alpha_2(|x|)$, it can select as $|x| + c$. Considering that for $\forall \mathbf{x} \in \mathbb{X}_f$, $V_f(\mathbf{x})$ is smaller than c, $V_f(\mathbf{x})$ is definitely smaller than $\alpha_2(|x|)$ designed above.
With both assumption 2.14(a) and 2.14(b) satisfied, the basic stability assumption is therefore satisfied.

(iv) **Assumption 2.17** (Weak controllability)
As described in [8], this assumption is weaker than a controllability assumption. It can be easily proved that

$$V_N^0(\mathbf{x}) \leq V_f(\mathbf{x}) \leq \alpha_2|\mathbf{x}|. \quad (32)$$

Therefore this is satisfied in this system as basic stability assumption has been proved beforehand.

(v) **Theorem 2.19** (Asymptotic stability of the origin)
According to **Theorem 2.19**, the equilibrium point is asymptotically stable because Assumption 2.2, 2.3, 2.14 and 2.17 are satisfied.

C. Proof of stability for system when $\kappa \neq 0$

The system becomes a time-varying when setting $\kappa \neq 0$. As described in the Section 2.4 [8], the **Theorem 2.39** proves the time-varying MPC, which needs to satisfy **Assumption 2.25**, **Assumption 2.26**, **Assumption 2.33** and **Assumption 2.37** in [8].

(i) **Assumption 2.25** (Continuity of system and cost; time-varying case)
Since both the stage cost $\ell(\mathbf{x}, \mathbf{u}, i)$ and the terminal cost $V_f(\mathbf{x}, i)$ are continuous as they are all in quadratic form. And when both state and control input are set to 0, both functions have a value of 0 at whatever time step, which means that the system satisfies $f(0,0,i) = 0$, $\ell(0,0,i) = 0$, $V_f(0,i) = 0$. This assumption is therefore satisfied.

(ii) **Assumption 2.26** (Properties of constraint sets; time-varying case)
Considering the state constraints and control input constraints discussed for the time-varying system, set $\mathbb{Z}$ is therefore closed. The terminal set $\mathbb{X}_f(i)$ is both compact and closed since this is a sub-level set. With both sets containing the origin point, which is the point where linearization is performed. This assumption is therefore satisfied.

(iii) **Assumption 2.33** (Basic stability assumption; time-varying case)
The **assumption 2.33**(a) shares the same proof as the **Assumption 2.14**(a) where we discussed earlier, the only difference between them is that the system is now time-varying. The only difference is that $A(i)$, $B(i)$, $K(i)$, $P(i)$ is now time-varying with respect to time. The proving process remains unchanged once substituting the time-invariant part as the time-varying parts. And it has been proved previously that the designed terminal set $\mathbb{X}_f$ is a control invariant set.
The **assumption 2.33**(b) can be proved with the same $\alpha_1|\mathbf{x}|$ and $\alpha_2|\mathbf{x}|$ used previously in proving the **Assumption 2.14**. The proving process is also the same once substituting the time-invariant part as the time-varying parts and the hyperparameter c used in time-varying case. With the two small requirements satisfied, this assumption is therefore satisfied.

(iv) **Assumption 2.37** (Uniform weak controllability)
Similarly, we can again show that

$$V_N^0(\mathbf{x}, i) \leq V_f(\mathbf{x}, i) \leq \alpha_2|\mathbf{x}|. \tag{33}$$

Therefore the assumption is satisfied.

(v) **Theorem 2.39** (Asymptotic stability of the origin: time-varying MPC)
According to **Theorem 2.39**, the origin is asymptotically stable because Assumption 2.2, 2.3, 2.14 and 2.17 are satisfied.

Therefore, it has been proved that both the linear MPC models are asymptotically stability around the equilibruim point along the reference trajectory.

IV. Numerical Simulations

In this section, we run several numerical simulations in Python Axes3D environment, which is shown as Fig. 3. And in the following part, we are going to discuss and prove the effectiveness of both the two MPC controllers. Moreover, we would also like to take the time-varying model as an example to discuss the different effects brought by changing various parameters, for example state stage cost and input stage cost, within this model.

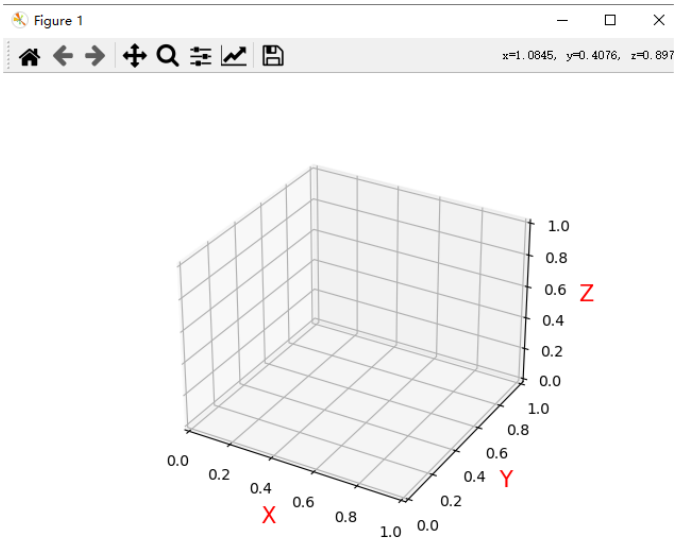

Fig. 3. Environments in Python used to run the simulations

A. Tracking for both MPC controllers

As illustrated previously in the report, our model is developed based on the application of a bevel-tip needle model. There are generally several different paths of injection in actual clinical applications. And these paths can be generated from needle models with different curvatures κ [9].

1) $\kappa = 0$: Specifically, when $\kappa = 0$, the dynamic model can be simplified into a linear model, whose motion is only in one dimension pz with a rotation angle γ. And the tracking task performed in the simulation environment introduced above is shown in Fig. 4.

It can be seen that the needle tip finally stops around the end point of the reference path. To better illustrate that the needle tip rotates to the final angle while climbing straightly upwards, the comparison between the needle tip's position and attitude is shown in Fig. 5.

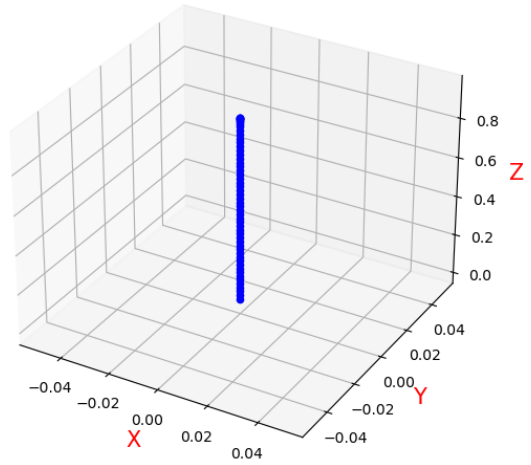

Fig. 4. Tracking task when $\kappa = 0$, where the blue line is the reference path and the dotted point is the position of needle tip each time step $dt = 0.1s$.

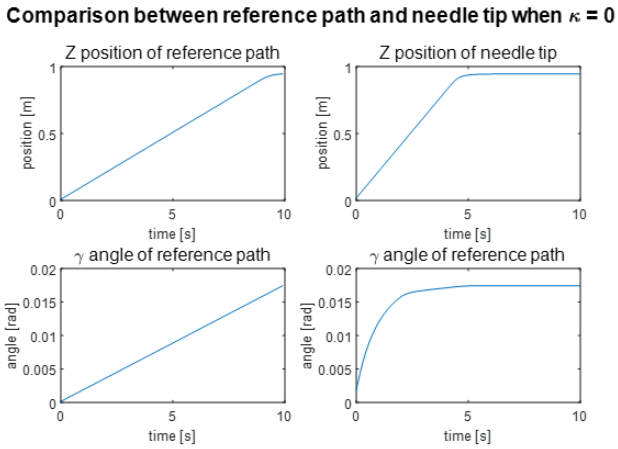

Fig. 5. The comparison between the needle tip and reference path when $\kappa = 0$ where the system is a time-invariant model.

2) $\kappa \neq 0$: According to Swaney, the patient will suffer form reduced tissue damage with a larger curvature κ [6]. And we believe that the model will be much more difficult to track for a higher κ, as the influence of rotating or stretching out to another orientation will be faster with a higher κ. Therefore, in our following experiment, we choose to set the curvature $\kappa = 1/150mm^{-1}$. To be mentioned, the simulator of the needle tip is using the original dynamics model without being linearized. The tracking result and comparison of our experiment is separately shown in Fig. 6 and Fig. 7.

To be mentioned, since that there are some difficulties for scipy package to calculate out the matrix $\mathbf{P}$ for some time step due to the value within matrix $\mathbf{A}$ is rather small, too much close to either 1 or 0. In the meantime, it is worth noting that the final destination of our reference path has a rather smaller x displacement comparing with y and z displacement.

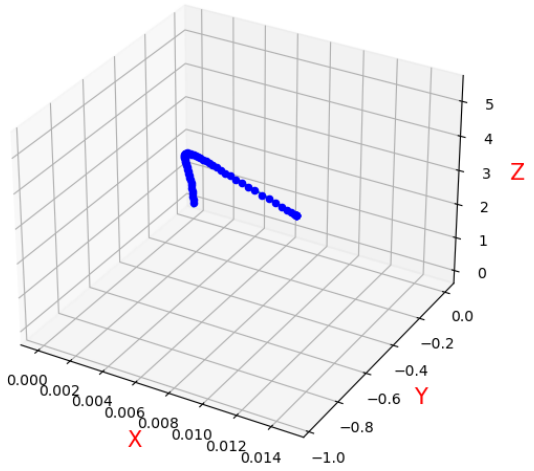

Fig. 6. Tracking task when $\kappa = 1/150mm^{-1}$, where the blue line is the reference path and the dotted point is the position of needle tip each time step $dt = 0.1s$.

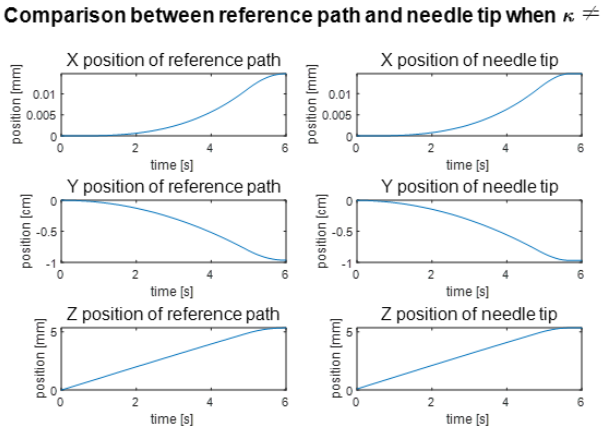

Fig. 7. The comparison between the needle tip and reference path when $\kappa \neq 0$ where the system is a time-invariant model.

Therefore, the matrix $\mathbf{A}$ and $\mathbf{B}$ is enlarged to 1000 and 100 separately for x and both y and z, which means that the final unit for x is in mm, and cm for y and z.

From the plots, we can see that our model successfully follow all the predetermined path rather precisely, and it tends to stay statistic near the end of the path around 5 seconds after starting.

In conclusion, we generate two different paths for either the time-invariant system or the time-varying system and further prove the effectiveness and precise of our designed MPC controller in this section. And for the following part in this report, we are going to take the time-varying system as an example to analyze the difference brought by the varying parameters. This is because it is the one path much more similar to the application in real life of a bevel-tip needle during surgery, where the needle tip goes along the vessels after inserting into the skin.

B. Different stage cost $\mathbf{Q}$

In this part, we compare the result of MPC performance brought by different choices of cost matrix $\mathbf{Q}$ for stages. In our model, we select the cost matrix as $\mathbf{Q} = diag(\begin{bmatrix}0.01 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1\end{bmatrix})$, based on our trials and understanding of the model. In this part, we tried few more stage cost matrix $\mathbf{Q}$, separately $\mathbf{Q} = 0.01\mathbf{I}$, $\mathbf{Q} = 0.1\mathbf{I}$, $\mathbf{Q} = \mathbf{I}$, $\mathbf{Q} = 10\mathbf{I}$. The result of the simulation with respect the position of reference path is shown in Fig .8.

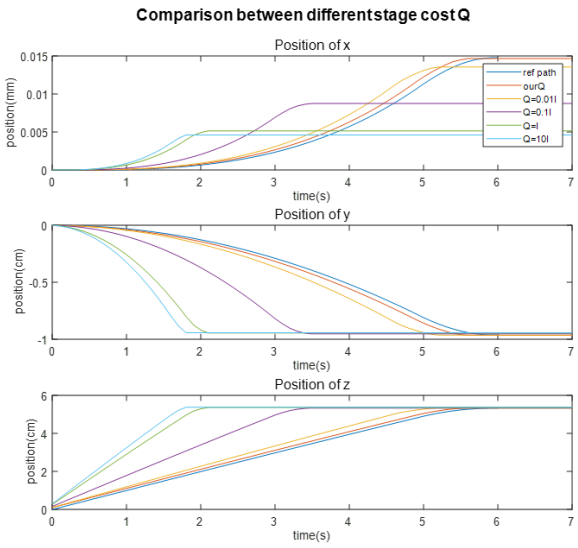

Fig. 8. The performance of position with different $\mathbf{Q}$ matrix

In the figures, It can be seen that the performance of our selected $\mathbf{Q}$ is the closet to the reference path among all these different stage cost matrix, especially for the precision in x direction. However, the precision in y direction has been sacrificed in this situation. The influence brought by the increasing value of matrix $\mathbf{Q}$ maybe due to the fact that larger control inputs are needed to lower the error as it can be seen from the figure that there is a larger velocity in this situation. We believe that in practical use precision in x really matters since that the y direction is along with the direction of vessel, and x is vertical to the direction of vessel. Therefore, this is why we select the matrix as our stage cost.

C. Different input stage cost $\mathbf{R}$

In this part, we perform the similar experiment as we do in the previous section, but this time the object is the input stage cost $\mathbf{R}$. In our model, we finally select $\mathbf{R} = diag(\begin{bmatrix}100 & 10\end{bmatrix})$ as our input stage cost matrix. The position deviation result between different cost matrix is shown in Fig. 9.

From the figures, we can tell that when giving a larger cost matrix to the input stage, the velocity will be greatly punished as the optimization process will give a smaller velocity v as the result. When the $\mathbf{R}$ is set to the smallest matrix, the velocity has nearly reached to the value of limitation. And it is worth noting that in this project, higher velocity will result in poor precision of position, this is because that all the angles rotated

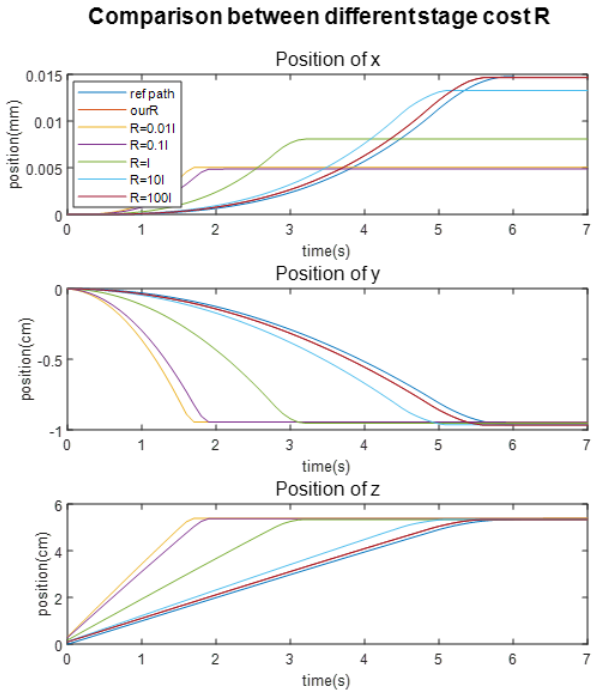

Fig. 9. The performance of position with different $\mathbf{R}$ matrix

in three dimensions, which is rather small at the beginning will have a great influence on the result. Since there is only a really small difference between setting $\mathbf{R} = diag(\begin{bmatrix}100 & 100\end{bmatrix})$ and our matrix, we choose this as our final selection. Moreover, it is obvious again that when setting the $\mathbf{R}$ to be the value used in our model, there is only slight difference for the position of x and y between the robot's model and reference path.

D. Different prediction horizon N

In this section, we perform experiments to see how the different prediction horizons (N) affect the result of our MPC model. As illustrated before, precision of the result is preferred for the needle tip to track the reference path. And hereby we separately select the prediction horizon as 8 and 20.

The comparison result between different prediction horizon for the needle's tip position in the world coordinate system is shown in Fig. 10. It is obvious that there are almost no difference for our time-varying MPC controller between the prediction horizons. This further proves that the linearized model has a rather good performance for the dynamic system described in this study. This also proves that the cost matrix $\mathbf{Q}$ and $\mathbf{R}$ is already optimized for our current model. We also found that there is a rapidly increased running time of the code when setting horizon $N = 20$. Therefore, we finally choose $N = 8$ as our parameter for prediction horizon.

V. Discussion

While implementing this project, we actually do meet with few problems would like to share. The first problem is whether there is a much easier way to linearize such a dynamic system.

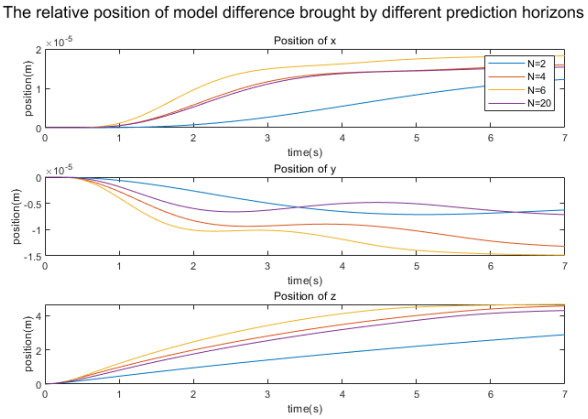

Fig. 10. The performance of position with different horizons. To better show that the result of different horizon share the same result, the plot of horizon $N = 20$ has the same as horizon $N = 8$.

As shown in Equation 14, the kinematic model we are using is of rather high non-linearity and there are lots of parameters in state of the robot used to determine another single state. The linearization method we used here is using Taylor expansion around the trajectory, and we would like to see whether there will be a much easier way in doing this. Moreover, the article we cited itself preserve its nonlinearity and use nonlinear mpc model when they perform their experiment [5]. And we are rather interested in the result of this nonlinear model.

The other question remains unsolved is that we are not sure about how to solve the reference state for our needle tip if there is only reference position given. This is also why the reference path used in this study for time-varying model is with a rather low speed. Since the orientation of the robot on a curve may be totally different from that of on a straight line especially when the velocity is high and the time between time step is long. Therefore, it is really hard to calculate the angle rotated and the reference speed out when the curve between two time steps is a circular curve.

In the future, we are going to explore the nonlinear mpc model of this dynamics sytem and also try to read more papers to find a way to calculate the reference state, including the attitude angles between two given positions. We think the application of model predictive control in the bevel-tip needle can definitely be well developed to help patients relieve pain and reduce the workload of medical staff in the future.

VI. Conclusion

In this paper, we successfully build a model predictive control model for the bevel-tip needle model to track the given path when both taking it as a normal straight needle and preserving its curvature. By illustrating the dynamic model and MPC design, we present our procedure to get to our final code. With the stability proof, we manage to prove the theory background for our linear MPC model, time-invariant when $\kappa = 0$ and time-varying when $\kappa \neq 0$. Finally, we use reference path to test the effectiveness of our MPC model. In the meantime, using different parameter sets yields that our solution is already optimal and effective to solve the problem.

REFERENCES

[1] H. Holm, J. K. Kristensen, S. N. Rasmussen, A. Northeved, and H. Barlebo, "Ultrasound as a guide in percutaneous puncture technique," *Ultrasonics*, vol. 10, no. 2, pp. 83–86, 1972.

[2] D. J. Kopacz and H. W. Allen, "Comparison of needle deviation during regional anesthetic techniques in a laboratory model," *Anesthesia & Analgesia*, vol. 81, no. 3, pp. 630–633, 1995.

[3] R. J. Webster III, J. S. Kim, N. J. Cowan, G. S. Chirikjian, and A. M. Okamura, "Nonholonomic modeling of needle steering," *The International Journal of Robotics Research*, vol. 25, no. 5-6, pp. 509–525, 2006.

[4] R. Secoli, F. Rodriguez *et al.*, "Experimental validation of curvature tracking with a programmable bevel-tip steerable needle," in *2018 International Symposium on Medical Robotics (ISMR)*. IEEE, 2018, pp. 1–6.

[5] B. Huo, X. Zhao, J. Han, and W. Xu, "Path-tracking control of bevel-tip needles using model predictive control," in *2016 IEEE 14th International Workshop on Advanced Motion Control (AMC)*. IEEE, 2016, pp. 197–202.

[6] P. J. Swaney, J. Burgner, H. B. Gilbert, and R. J. Webster, "A flexure-based steerable needle: high curvature with reduced tissue damage," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 4, pp. 906–909, 2012.

[7] J. C. Chou and M. Kamel, "Finding the position and orientation of a sensor on a robot manipulator using quaternions," *The international journal of robotics research*, vol. 10, no. 3, pp. 240–254, 1991.

[8] J. Rawlings and D. Mayne, *Model Predictive Control: Theory and Design*. Nob Hill Publishing, 2008.

[9] R. Alterovitz, K. Goldberg, and A. Okamura, "Planning for steerable bevel-tip needle insertion through 2d soft tissue with obstacles," in *Proceedings of the 2005 IEEE international conference on robotics and automation*. IEEE, 2005, pp. 1640–1645.