Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnline CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 56-3 (2023) 583–588

A Data-Driven Approach to Motion Planning and Optimal Control of Medical Nanorobots with Dynamic Window Approach ⋆

Prahlad Pandav * **Juan Ren** **

* *Iowa State University, Ames, IA 60030 USA (e-mail: ppandav@iastate.edu).*

** *Iowa State University, Ames, IA 50010 USA (corresponding author, e-mail: juanren@iastate.edu)*

Abstract: Targeted drug delivery via nanorobots has become an area of prominent research. However, since such robots are microns in size (and in the future, nanometers in size), implementing control laws to make sure the nanorobot does not deviate from a desired trajectory becomes exceedingly difficult. Building upon the existing nonlinear dynamic models of a nanorobot, this work proposes a data-driven method for optimal path planning and control of nanorobots. Specifically, the collision avoidance strategy, Dynamic Window Approach (DWA) implemented with linear quadratic regulator (LQR) control, is employed to simulate a randomly chosen nanorobot moving through the pulmonary artery against blood flow. The nanorobot is simulated as though it is being guided from the initial site to the target site via an induced magnetic field gradient generated by an MRI. It is shown that when full-state feedback control is implemented for nanorobot control, the LQR controller is able to quickly bring the nanorobot to desired states.

Keywords: Dynamic window approach, dynamic mode decomposition, LQR control, nanorobotics

1. INTRODUCTION

In recent medical advancements, targeted drug delivery via nanorobots has become an area of prominent research (Hu et al. (2020)). From developments in nanorobot design to propulsion methods toward a target site, nanorobots are becoming a viable option for many delicate surgical and other therapeutic procedures, making the delivery of certain drugs efficient and significantly safer than traditional medical techniques (Luo et al. (2018); Li et al. (2017)).

Recent research in medical nanorobotics ranges from propulsion methods to nanorobot design for efficient delivery (Hu et al. (2020)). As per Hu et al. (2020), propulsion methods include, but are not limited to, the use of electric and magnetic fields, light energy, ultrasound energy, and chemical energy. In recent developments, biological fuel has been used as an additional source of propulsion toward a target site. For instance, Urso and Pumera (2022) utilizes DNA as a fuel source to propel biorobots toward cancer cell-occupied target sites. The design of nanorobots is also critical as the geometrical design can dictate how efficiently the nanorobot can 'swim' (Giri et al. (2021)). For instance, Gao et al. (2014) utilizes magnetic helical swimmers, which are directly fabricated from the helical xylem vessels of various vascular plants. The rotation of the helical swimmers, which is induced by an applied external magnetic field, allows the nanorobots to propel toward the target site. However, magnetic-driven nanorobots are mostly preferred as this actuation enables long-range, fast, and precise actuation of single or many robots in diverse bodily environments. Meanwhile, magnetic actuation has unique potential for medical applications of microrobots inside nontransparent tissues at high penetration depths (Sitti and Wiersma (2020)). Thus, this study focuses on MRI-driven nanorobots.

One key factor in targeted drug delivery is path design and existing path planning algorithms in robotics provide optimal solutions to path designs. For instance, the A* search algorithms attempt to traverse through a graph, calculate weights and distances between each node, and determine which nodal direction is optimal (least cost) to reach a target (Russell and Norvig (2021); Yao et al. (2010)). In contrast, RRT (Petit and Desbiens (2021)), RRT* (Adiyatov and Varol (2017)), and Dynamic Window Approach (DWA) (Fox et al. (1997); Sakai et al. (2018)) algorithms are forms of collision avoidance policies. Among these approaches, DWA is a physics-informed path planner that computes a feasible path based on the robot's current state (Fox et al. (1997); Liu et al. (2021)) as it directly takes into account the robot dynamics (position, velocity, acceleration, yaw angle, and yaw rate) and computes an optimal path. Due to the microscopic nature of the nanorobot, considering nanorobot dynamics is vital, and thus, DWA is ideal for our study of micro/nano-robot control. Moreover, since such robots are extremely small, integration of the path planning approach with feedback control is necessary to ensure robust tracking of the robot. Therefore, in this study, we propose integrating DWA with a full-state feedback controller for motion planning of MRI-driven nanorobots.

This paper introduces a new method for targeted drug delivery using nanorobots, in which an optimal path can be configured and followed along using DWA integrated with Linear Quadratic Regulator (LQR). The research presented in this paper is building upon the nonlinear dynamics of the nanorobots

⋆ This work is supported by the Iowa State University Graduate Assistantship funds.

10.1016/j.ifacol.2023.12.087

(Arcese et al. (2009)), based on which a linearized state equation has been derived using the Dynamic Mode Decomposition with Control (DMDc) algorithm and then implemented into the DWA algorithm integrated with LQR control. This approach has been implemented in simulation, and the results demonstrate that the controller can effectively track the nanorobot.

2. METHODS

The assumptions we made at the start of this research are as follows:

- The nanorobot is spherical.
- The nanorobot will travel against blood flow.
- Blood compressibility is ignored (Thomas and Sumam (2016)).
- Elasticity and deformation effects (Calandrini et al. (2018)) of the artery wall are ignored.
- Blood flow is laminar, steady, and irrotational in the pulmonary artery with a freestream blood velocity of u_∞ (Gabe et al. (1969)).
- The nanorobot is assumed to be guided by an MRI towards the target site via magnetic field gradients. The magnetic field gradients are the control inputs in the system.
- Force exerted on the nanorobot due to external magnetic field must be greater than the drag force the nanorobot experiences, and thus must be accelerating.

2.1 Nanorobot System Dynamics

Equations of motion of the nanorobot can be derived by free body diagram analysis (Arcese et al. (2009)), as shown in Fig. 1.

By free body diagram analysis, the drag force is given as

$$F_d = -\frac{1}{2}\rho(u_r - u_\infty)^2 A C_d \tag{1}$$

where $u_r - u_\infty$ is the nanorobot's relative velocity with respect to the freestream blood velocity, A is the frontal area of the nanorobot, and C_d is the drag coefficient. In a laminar flow regime with a large Reynold's Number, the drag coefficient for a spherical object is formulated as

$$C_d = \frac{24}{Re} + \frac{6}{1+\sqrt{Re}} + 0.4 \tag{2}$$

where Re is the Reynold's Number.

For the nanorobot to maintain its course toward a specified target site, the force due to the external magnetic field must be greater than the drag force. The force due to the external magnetic field is

$$F_m = \mu_0 V_m (\overrightarrow{M} \cdot \overrightarrow{\nabla}) \cdot \overrightarrow{B} \tag{3}$$

where $\mu_0 = 4 \cdot \pi \times 10^{-7}\,\mathrm{T \cdot m/A}$ is the free space permeability, V_m is the magnetic core volume, M is the magnetization, and B is the magnetic field generated by the MRI. The weight of the nanorobot is: $W_a = (\rho_r - \rho_b)gV$, where V is the volume of the nanorobot and ρ_r is the density of the nanorobot.

The equations of motion in the $\hat{i}$ and $\hat{k}$ directions (see Fig. 2) can then be derived as (Arcese et al. (2009))

$$m\ddot{x} = \overrightarrow{F}_{d_x} + \overrightarrow{F}_{m_x} \tag{4}$$

$$m\ddot{z} = \overrightarrow{F}_{d_z} + \overrightarrow{F}_{m_z} + \overrightarrow{W}_a \tag{5}$$

It should be noted that x and z are the projections on the unit vectors $\hat{i}$ and $\hat{k}$, respectively. Since the dynamics in both directions are similar, the control law formulations for dynamics in either direction would also be similar. Therefore, we focus on the nanorobot translation motion in the $\hat{i}$ direction. With $\hat{k}$ direction dynamics negated, the following equation of motion can be derived

$$\ddot{x} = a_1(\dot{x} - u_\infty) + a_2(\dot{x} - u_\infty)^2 + a_3 \frac{(\dot{x} - u_\infty)^2}{1 + \alpha\sqrt{\dot{x} - u_\infty}} + a_4 u \tag{6}$$

Further information on nanorobot parameters a_i and α $[\sqrt{\mathrm{s}}\,\mathrm{m}^{-1}]$ can be found in the work by Arcese et al. (2009). The control input u is the the magnetic field gradient input $u = ||\overrightarrow{\nabla}\overrightarrow{B}||$ $[\mathrm{Tm}^{-1}]$. Additionally, it is important to note that robot parameters a_1 $[\mathrm{m\,s}^{-1}]$, a_2 $[\mathrm{m}^{-1}]$, and a_3 $[\mathrm{m}^{-1}]$ are functions of the yaw angle θ.

The yaw (heading) angle is the angle of the nanorobot's direction of travel with respect to the target site. As the DWA motion planner seeks to compute and generate smooth, curved trajectories, it is imperative to consider the yaw dynamics due to the yaw angle constantly changing along the robot's path.

We set the states x_1 and x_2 to be the position and velocity of the nanorobot in the $\hat{i}$ direction, and x_3 and x_4 to be the yaw angle and yaw rate of the nanorobot. The yaw dynamics can be derived by determining the sum of all moments acting on the nanorobot about its start position (origin) and coupling it with the resultant angular acceleration. Assuming the nanorobot's starting point is at the origin, the sum of all moments about the origin is $\Sigma M_O = (F_m - F_d)x_r \sin(x_3)$. Coupling the sum of all moments with the resultant angular acceleration, $M = I\ddot{\theta}$, the robot's yaw dynamics can be derived. Note that $\dot{x_4} = \ddot{\theta}$ and

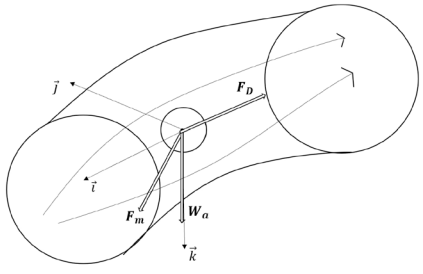

Fig. 1. Free body diagram of the nanorobot in a fully-developed laminar flow regime.

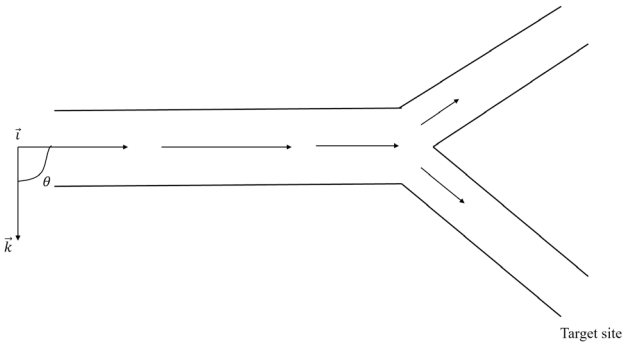

Fig. 2. Defined coordinate system with respect to a selected blood vessel.

I is the robot's moment of inertia. Therefore, the nonlinear state space model is then formulated as

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = f_2(x_2) + \bar{u}_1 \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = \frac{F}{I}\dot{x}_r t \sin(x_3) \\ y = [x_1\ x_3]^T \end{cases} \tag{7}$$

where $F = F_m - F_d$. Note x_1 and x_3 are measurable by MRI tracking. Also, $f_2(x_2) + \bar{u}_1 = a_1 x_2 + a_2(x_2^2 - 2u_\infty x_2) + a_3 \frac{(x_2-u_\infty)^2}{1+\alpha\sqrt{x_2-u_\infty}} + a_2 u_\infty^2 - a_1 u_\infty + a_4 u$.

2.2 Linearization of the Nanorobot System

For the nonlinear system shown in (7), traditional linearization techniques (e.g., linearization via Taylor series) may lead to poor control performance as the equilibrium points x_e may be unreachable or unstable regions. In this work, we propose a data-driven approach—the Dynamic Mode Decomposition with Control (DMDc) algorithm (Proctor et al. (2016))—to nanorobot system linearization.

The DMD algorithm seeks to reduce the high dimensionality of a nonlinear system and maps it to a linearized space such that the linearized model, $x(k+1) = Ax(k)$, captures the underlying dynamics of the system with high accuracy; this is derived from multidimensional spatiotemporal timeseries datasets. With DMDc, linear operators are computed by least squares regression, which then allows for computing the state and actuation matrices.

The datasets (i.e., the *snapshots*) contain the time history of the state vector, $X = [x_1, x_2, x_3, x_4]^T$, which can be obtained by numerically solving the set of stiff differential equations in Eq. (7) with a sample time dt. The snapshots of the state vector X_i are then merged into a large, high dimensional matrix as follows:

$$\mathbb{X} = [X_1, X_2, ...X_{m-1}], \quad \text{and} \quad \mathbb{X}' = [X_2, X_3, ...X_m], \tag{8}$$

where $\mathbb{X}' \in \mathbb{R}^{m\times 4}$ is the concatenated dataset $\mathbb{X} \in \mathbb{R}^{m\times 4}$ shifted by one sample step and m is the number of snapshots obtained. In addition, a large matrix Υ comprised of MRI control input measurements must also be determined. Similarly, $\Upsilon = [u_1, u_2, ..., u_{m-1}]$. It is worth mentioning that for this research, control input constraints were imposed as $[-u_{max},\ u_{max}]$, where $u_{max} = 0.045\ \mathrm{T\,m^{-1}}$.

The state matrix A can thus be approximated as $A = \mathbb{X}'\mathbb{X}^\dagger$, where $\mathbb{X}^\dagger$ denotes the Moore-Penrose pseudoinverse of $\mathbb{X}$. However, this computation is inefficient due to the high dimensionality of $\mathbb{X}$. Therefore, we compute the singular value decomposition (SVD) of $\mathbb{X}$ and commit rank-r_{SVD} truncation with $r_{SVD} = 4$. By rank-r_{SVD} truncation, the high dimensionality of $\mathbb{X}$ is reduced, which is essentially what the DMD algorithm aims to do. Additionally, the control space Ω must also be truncated by a higher rank-$p_{SVD} = 10$. Therefore, a linearized, data-driven based state equation can be written as

$$\mathbb{X}' = A\mathbb{X} + B\Upsilon \Longrightarrow \mathbb{X}' = [A\ B]\begin{bmatrix}\mathbb{X}\\ \Upsilon\end{bmatrix} = G\Omega, \quad y = C\mathbb{X} \tag{9}$$

where $G = [A\ B]$ and $\Omega = [\mathbb{X}\ \Upsilon]^T$ (Proctor et al. (2016)).

If SVD of $\mathbb{X}'$ and Ω, with rank-r_{SVD} and rank-p_{SVD}, truncations are performed, the linearized state and control input mappings can be rewritten as

$$\begin{aligned} A &\approx \hat{U}\mathbb{X}'\tilde{V}\tilde{\Sigma}^{-1}\tilde{U}_1\hat{U} \\ B &\approx \hat{U}\mathbb{X}'\tilde{V}\tilde{\Sigma}^{-1}\tilde{U}_2 \end{aligned} \tag{10}$$

where terms with $\hat{\cdot}$ and $\tilde{\cdot}$ are derived from computing the SVD of $\mathbb{X}'$ and Ω, respectively, and $\tilde{U} = [\tilde{U}_1\ \tilde{U}_2]^T$.

Examining the DMD (Koopman) modes and eigenvalues of A can provide insight into the behavior and stability of the linearized system in (9). DMD modes depict the spatiotemporal patterns within a dynamical system, with each mode comprised of spatial and temporal patterns corresponding to each state. Modal analysis can be performed by computing the DMD modes Φ as

$$\Phi = \mathbb{X}'V\Sigma^{-1}\tilde{U}_1\hat{U}W \tag{11}$$

where W is the eigenvector of the estimated state matrix (Proctor et al. (2016)) .

2.3 Dynamic Window Approach

In this work, possible obstacles that are considered a nanorobot might encounter include blood cells, tissues, and collisions with blood vessel walls. It has been suggested that admissible velocities can be determined by the objective function, which is a function of the robot's velocity, distance to the next nearest obstacle, and progress towards its target (Fox et al. (1997)). Based on these parameters, the DWA algorithm computes a search space that allows the planner to increase or decrease the robot's velocity, as required, within the proximity of an obstacle or target site. The admissible velocities, V_a, of a given robot within a dynamic window are provided as

$$\begin{aligned} V_a = \{v, \omega | \min(v \le \sqrt{2 \cdot dist(v,\omega)} \cdot \dot{v_b}, \omega \le \\ \sqrt{2 \cdot dist(v,\omega)} \cdot \dot{\omega_b}\}, \end{aligned} \tag{12}$$

where v and ω are the linear and angular velocities of the robot along its trajectory, $dist(v,\omega)$ is the distance to the nearest oncoming obstacle, and $\dot{v_b}$ and $\dot{\omega_b}$ are the accelerations required for breakage (Fox et al. (1997)).

Within a certain time interval t in the narrowed down search space, the robot will travel with actual velocities v_a and ω_a. Thus, the dynamic window can be represented as

$$V_d = \{(v,\omega) | \min(v\varepsilon[v_a \pm \dot{v}\cdot t], \omega\varepsilon[\omega_a \pm \dot{\omega}\cdot t])\} \tag{13}$$

By introducing boundary conditions to the search space, we obtain a resulting search space V_r Fox et al. (1997). Within V_r is a set of feasible velocities V_s, which can be expressed as

$$V_r = V_s \cap V_a \cap V_d. \tag{14}$$

Finally, the optimal trajectory is computed by minimizing the objective cost function, $G(v,\omega)$, which is minimized to determine optimal linear and angular velocities to the robot's trajectory. The objective function is given by

$$\begin{aligned} G(x_2,x_4) = \sigma(\alpha \cdot heading(x_2,x_4) + \\ \beta \cdot dist(x_2,x_4) + \gamma \cdot velocity(x_2,x_4)) \end{aligned} \tag{15}$$

where α, β, and γ are the weighted target, obstacle, and velocity cost parameters, respectively, and $heading(x_2,x_4)$ is the instantaneous yaw angle with respect to the target heading. With the above considerations, the next subsection explores optimal motion planning of the nanorobot using DWA with Linear Quadratic Regulator (LQR) controller application.

2.4 LQR-DWA Approach

LQR control is integrated into the open source DWA algorithm (Sakai et al. (2018)) to ensure the nanorobot system can operate at minimal cost and stability. The following is the LQR cost

function that minimizes the sum of all state errors and the control input,

$$J_{LQR} = \sum_{k=0}^{\infty} (\mathbb{X}_k^T Q \mathbb{X}_k + u_k^T R u) \tag{16}$$

where Q and R are the state and input cost weighted matrices, respectively, and $\mathbb{X}$ and u are the DMD linearized state and control input vectors of the nanorobot dynamic system. The stabilizing solution S can be found by solving the discrete time algebraic Riccati Equation (DARE) and is given by

$$S = A^T SA - (A^T SB)(R + B^T SB)^{-1}(B^T SA) + Q \tag{17}$$

Additionally, the optimal LQR gain matrix K and the LQR cost is computed by

$$K = R^{-1} B^T S \tag{18}$$

thus providing with the optimal policy

$$u = -K\mathbb{X} \tag{19}$$

For the linearized system in (9), the Q and R matrices were chosen to be $diag([100,\ 1,\ 100,\ 1])$ and 0.01, respectively. With Python's Control Systems library, the discrete time LQR problem becomes

$$[K,\ S] = dLQR(A,\ B,\ Q,\ R) \tag{20}$$

which allows to derive the closed loop dynamics $A_C = A - BK$. Since DWA has its own cost function, a balance between LQR and DWA costs must be maintained. A common technique to sum multiple cost functions is to sum them with a weighting factor w (Adachi et al. (2019)). Therefore, the total cost can then be computed as

$$J = wG(x_2, x_4) + (1 - w)J_{LQR} \tag{21}$$

With (21), the DWA algorithm can compute trade-offs between the two cost functions, determining which objective is prioritized; if w is large, the DWA cost is prioritized and vice versa. Here, we set $w = 0.2$. Section III presents the results of LQR with DMDc control implementation in the DWA algorithm.

3. RESULTS AND DISCUSSION

With the open source DWA algorithm developed by Sakai et al. (2018), LQR control was implemented and tested. The nanorobot is simulated as if it were being guided by an MRI through a section of the pulmonary artery.

3.1 Software Methodologies

For linearization of the nonlinear nanorobot dynamics, the set of ordinary differential equations (ODE) in (7) was numerically solved with *ode23s* as the ODEs are strongly coupled and stiff. The numerical solutions for each state and control input were collected and concatenated into the high dimensional matrices X_i and Υ, as shown in (8); by performing respective SVD computations in (9), DMD modes of the system were obtained. The state and actuation matrices were then estimated by least square regression with the PyKoopman library.

Throughout this work, MATLAB R2022a and PyCharm 2020.3.3 with Python 3.8.10 were utilized. For simulating the control of the nanorobot, MATLAB's Control Systems library was used extensively to simulate the open and closed loop responses of the nanorobot system. Additional factors contributing to nanorobot dynamics, such as hydrodynamic and magnetic forces, were accounted for in the motion planner.

Due to limitations posed by the Matplotlib animation library, robots with microscopic radii (Table 1) cannot be simulated in the DWA planner. To conserve blood flow conditions in the path planner and match the peak Reynold's number of $Re = 3400$ in the pulmonary artery (Capuano et al. (2019)), we apply a scaling factor of 140.68 to obtain an equivalent radius of $r_{DWA} = 0.0422$ cm, which is within range of acceptable robot radii.

3.2 Closed Loop Position & Yaw LQR Control

After obtaining the open-loop system dynamics in (9) via DMDc, we find that the controllability and observability matrices, Q_C and Q_O, are full rank, and conditions for full-state feedback control are satisfied. Considering the open and closed loop eigenvalues λ_i of the state matrix, one finds that the magnitudes of all eigenvalues are significantly less than 1, and the eigenvalues lie inside the unit circle, making the open and closed loop systems asymptotically stable. With parameters presented in Table 1, the step response of the linearized system is shown in Figure 3.

As can be seen in the step response, the position state is able to respond to the control input quicker than the yaw state. This is expected as the sole purpose of the MRI-generated gradients is to propel the nanorobot toward the target site. By introducing LQR control, as in Figure 4, we found that the transient response of the closed-loop system step response is significantly improved. Here, the LQR controller was able to bring the position and yaw states to the steady state much faster (e.g., in terms of the settling time: the position state was able to reach a unity steady-state within 0.01 s.) compared to the open-loop system, implying that LQR control of the nanorobot effectively achieves the desired response.

Figure 5(a) depicts the tracking error $e = x_r - x_l$ (Arcese et al. (2009)) of the longitudinal position for sinusoidal reference with LQR control, with $x_r = 1 + \sin(0.5\pi t)$cm. From Figure 5(a), it is shown that the closed-loop system is able to follow the desired trajectory with good accuracy, i.e., the peak-peak tracking error is less that 2% with respect to the reference peak-peak amplitude (see Figure 5). As shown in Figure 5(b),

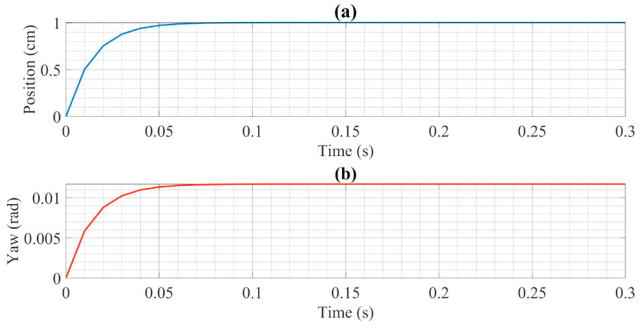

Fig. 3. Open-loop position (a) and yaw (b) step response.

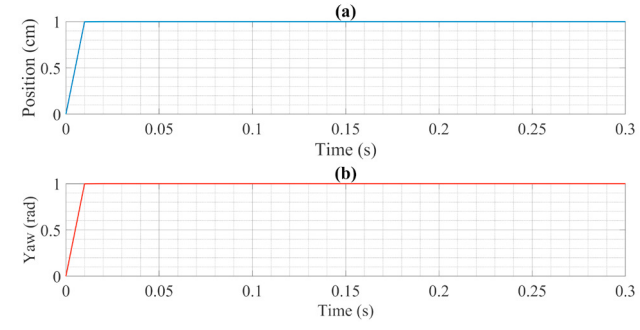

Fig. 4. Closed-loop position (a) and yaw (b) step response.

the maximum error is 0.0401 cm and the root mean squared error is (RMSE) of 0.0281 cm. With such a low error, we can conclude that the LQR controller can effectively track the movement of the nanorobot. The error can be further reduced if the sampling time used for the DMD approach is lower. However, this also increases the computation requirement for a real-time processor.

3.3 LQR-Dynamic Window Approach Simulation

The nanorobot and fluid parameters defined previously (e.g., in Arcese et al. (2009)) are provided in Table 1.

Table 1. Physical parameters of the nanorobot and blood.

Radius	r	300×10^{-6} m
Blood viscosity	η	15×10^{-3} Pa · s
Blood density	ρ_b	8000 kg/m^3
Robot density	ρ_r	1060 kg/m^3
Magnetization	M	1.950×10^6 A/m
Moment of inertia	I	3.26×10^{-14} kg · m^2

The parameters in Table 2 have been used in simulation in order to reduce the search space and establish a dynamic window: velocity, yaw rate, acceleration, velocity and yaw rate resolutions, time step, goal cost, velocity cost, and obstacle cost. Figure 6 shows the path of the nanorobot. With LQR control implemented, the nanorobot was able to travel along the artery based on the DWA-designed trajectory. The green dot is the nanorobot's starting point, the red line is the path computed based on the nanorobot's current state, the blue dot is the spherical nanorobot, and the target site is the location where the nanorobot stopped. The randomly generated black dots in the artery are the obstacles such as red blood cells or other particles travelling with the flow of blood. Note that the two horizontal, continuous black dots in 6 act as the arterial

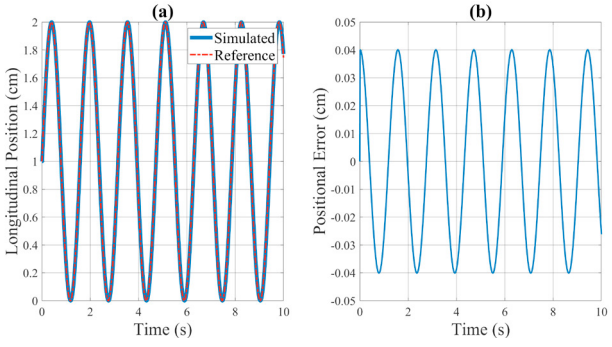

Fig. 5. (a) Sinusoidal longitudinal trajectory tracking; (b) Tracking error.

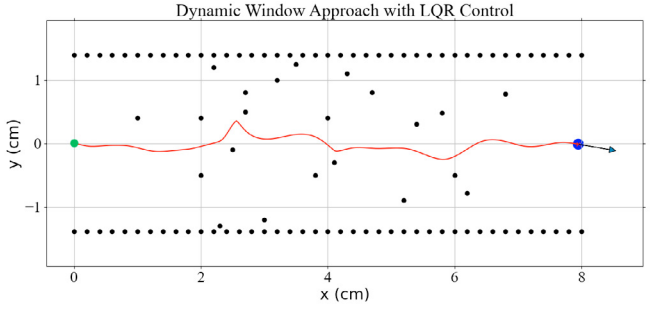

Fig. 6. Path of nanorobot after LQR control implementation.

walls of a selected section of the pulmonary artery. As can be seen, the nanorobot is able to travel to the target site without collisions with the obstacles. A sharp curve around $x = 2.3$ cm is noticed on the path, which indicates that the motion planner had to quickly decrease its speed and bring the nanorobot to a halt. After determining a feasible path, the robot quickly accelerates and corrects its yaw angle in the direction of the target site. Additionally, the nanorobot responds quickly to oncoming obstacles due to the added robustness from LQR control.

Table 2. Dynamic Window Approach Parameters for Nanorobot

dt	0.01 s
v_{max}	0.35 m s^{-1}
v_{min}	-0.5 m s^{-1}
x_0	$[0 \ -0.57 \ \frac{\pi}{8} \ 0]^T$
Maximum yaw rate	$\frac{40\pi}{180}$ rad/s
Maximum acceleration	0.2 m/s^2
Velocity resolution	0.01 m/s
Yaw rate resolution	$\frac{0.1\pi}{180}$ rad/s
Maximum yaw acceleration	$\frac{40\pi}{180}$ rad/s^2
Goal cost	0.15
Velocity cost	1.0
Obstacle cost	1.0

4. CONCLUSION AND FUTURE WORK

This study introduces a baseline framework for linear control of magnetically-driven nanorobots with linear quadratic regulator control to the Dynamic Window Approach motion planner, and is used to determine and simulate an optimal path for an MRI-guided nanorobot to follow along in a bodily environment. The nonlinear dynamics of the nanorobot were linearized by obtaining a linear model with Dynamic Mode Decomposition. Introducing LQR control to the linearized system and the DWA path planner made the control of the nanorobot more robust and responsive, as demonstrated by the simulation the results.

In future studies, we aim to consider the case where the nanorobot's motion is affected by pressure differences and blood compressibility. This will allow for obtaining a linearized dynamical model that will accurately reflect the actual motion of the nanorobot against blood flow, and we expect control performances to improve if controllers, such as H_∞, or adaptive controllers are implemented. In reality, obstacles in the bloodstream move with the flow of blood, and, therefore, it is necessary to observe how the nanorobot will respond to dynamic obstacles with LQR control.

REFERENCES

Adachi, D., Tsujimoto, N., Akashi, R., Todo, S., and Tsuneyuki, S. (2019). Search for common minima in joint optimization of multiple cost functions. *Computer Physics Communications*, 241, 92–97.

Adiyatov, O. and Varol, H.A. (2017). A novel rrt*-based algorithm for motion planning in dynamic environments. In *2017 IEEE International Conference on Mechatronics and Automation (ICMA)*, 1416–1421. IEEE.

Arcese, L., Fruchard, M., and Ferreira, A. (2009). Nonlinear modeling and robust controller-observer for a magnetic microrobot in a fluidic environment using mri gradients. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 534–539. IEEE.

Calandrini, S., Capodaglio, G., and Aulisa, E. (2018). Magnetic drug targeting simulations in blood flows with fluid-structure

interaction. *International journal for numerical methods in biomedical engineering*, 34(4), e2954.

Capuano, F., Loke, Y.H., and Balaras, E. (2019). Blood flow dynamics at the pulmonary artery bifurcation. *Fluids*, 4(4), 190.

Fox, D., Burgard, W., and Thrun, S. (1997). The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*, 4(1), 23–33.

Gabe, I.T., GAULT, J.H., ROSS JR, J., MASON, D.T., MILLS, C.J., SCHILLINGFORD, J.P., and Braunwald, E. (1969). Measurement of instantaneous blood flow velocity and pressure in conscious man with a catheter-tip velocity probe. *Circulation*, 40(5), 603–614.

Gao, W., Feng, X., Pei, A., Kane, C.R., Tam, R., Hennessy, C., and Wang, J. (2014). Bioinspired helical microswimmers based on vascular plants. *Nano letters*, 14(1), 305–310.

Giri, G., Maddahi, Y., and Zareinia, K. (2021). A brief review on challenges in design and development of nanorobots for medical applications. *Applied Sciences*, 11(21), 10385.

Hu, M., Ge, X., Chen, X., Mao, W., Qian, X., and Yuan, W.E. (2020). Micro/nanorobot: A promising targeted drug delivery system. *Pharmaceutics*, 12(7), 665.

Li, J., de Ávila, B.E.F., Gao, W., Zhang, L., and Wang, J. (2017). Micro/nanorobots for biomedicine: Delivery, surgery, sensing, and detoxification. *Science robotics*.

Liu, L., Yao, J., He, D., Chen, J., Huang, J., Xu, H., Wang, B., and Guo, J. (2021). Global dynamic path planning fusion algorithm combining jump-a* algorithm and dynamic window approach. *IEEE Access*, 9, 19632–19638.

Luo, M., Feng, Y., Wang, T., and Guan, J. (2018). Micro-/nanorobots at work in active drug delivery. *Advanced Functional Materials*, 28(25), 1706100.

Petit, L. and Desbiens, A.L. (2021). Rrt-rope: A deterministic shortening approach for fast near-optimal path planning in large-scale uncluttered 3d environments. In *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 1111–1118. IEEE.

Proctor, J.L., Brunton, S.L., and Kutz, J.N. (2016). Dynamic mode decomposition with control. *SIAM Journal on Applied Dynamical Systems*, 15(1), 142–161.

Russell, S. and Norvig, P. (2021). Artificial intelligence: a modern approach, global edition 4th. *Foundations*, 19, 23.

Sakai, A., Ingram, D., Dinius, J., Chawla, K., Raffin, A., and Paques, A. (2018). Pythonrobotics: a python code collection of robotics algorithms. doi:10.48550/ARXIV.1808.10703. URL https://arxiv.org/abs/1808.10703.

Sitti, M. and Wiersma, D.S. (2020). Pros and cons: Magnetic versus optical microrobots. *Advanced Materials*, 32(20), 1906766.

Thomas, B. and Sumam, K. (2016). Blood flow in human arterial system-a review. *Procedia Technology*, 24, 339–346.

Urso, M. and Pumera, M. (2022). Micro-and nanorobots meet dna. *Advanced Functional Materials*, 2200711.

Yao, J., Lin, C., Xie, X., Wang, A.J., and Hung, C.C. (2010). Path planning for virtual human motion using improved a* star algorithm. In *2010 Seventh international conference on information technology: new generations*, 1154–1158. IEEE.