

Model Predictive Control-Based Path-Following for Tail-Actuated Robotic Fish

Maria L. Castaño¹

Smart Microsystems Lab,
Department of Electrical and
Computer Engineering,
Michigan State University,
East Lansing, MI 48824
e-mail: castanom@msu.edu

Xiaobo Tan

Smart Microsystems Lab,
Department of Electrical and
Computer Engineering,
Michigan State University,
East Lansing, MI 48824
e-mail: xbtan@msu.edu

There has been an increasing interest in the use of autonomous underwater robots to monitor freshwater and marine environments. In particular, robots that propel and maneuver themselves like fish, often known as robotic fish, have emerged as mobile sensing platforms for aquatic environments. Highly nonlinear and often under-actuated dynamics of robotic fish present significant challenges in control of these robots. In this work, we propose a nonlinear model predictive control (NMPC) approach to path-following of a tail-actuated robotic fish that accommodates the nonlinear dynamics and actuation constraints while minimizing the control effort. Considering the cyclic nature of tail actuation, the control design is based on an averaged dynamic model, where the hydrodynamic force generated by tail beating is captured using Lighthill's large-amplitude elongated-body theory. A computationally efficient approach is developed to identify the model parameters based on the measured swimming and turning data for the robot. With the tail beat frequency fixed, the bias and amplitude of the tail oscillation are treated as physical variables to be manipulated, which are related to the control inputs via a nonlinear map. A control projection method is introduced to accommodate the sector-shaped constraints of the control inputs while minimizing the optimization complexity in solving the NMPC problem. Both simulation and experimental results support the efficacy of the proposed approach. In particular, the advantages of the control projection method are shown via comparison with alternative approaches.

[DOI: 10.1115/1.4043152]

1 Introduction

Aquatic ecosystem sustainability is often at risk due to the increase of potential threats, such as oil spills, invasive species, and industrial and household waste. As a result, monitoring and understanding aquatic environments has become essential to ensuring the longevity of aquatic ecosystems and for securing water resources. In recent years, a type of aquatic robots that mimics the movement of fish (Fig. 1) has emerged as an attractive choice for the aforementioned applications. These robots have various actuation mechanisms, from oscillating caudal or pectoral fins to undulation of the entire body, and like fish, they are able to attain high maneuverability [1].

To be suitable for monitoring such ecosystems, these robots need to be able to sustain long field-operation time, and it is, thus, crucial for them to be highly energy-efficient. The latter makes optimal control an important problem for robotic fish. Much of the work done for robotic fish has been in robot development [2–11] and modeling [12–18]. There has also been extensive work on motion control of robotic fish, which has mainly been focused on the generation of fish-like swimming gaits, and on control to drive the robot to achieve a desired motion. In the case of swimming gait generation, several kinematics and dynamics-based schemes [19–27], as well as bioinspired approaches, such as central pattern generators [28–33], have been used to produce fish-like swimming. However, these approaches are typically open-loop in nature. Although some works have examined trajectory-tracking or stabilization problems [12,34], they have mainly been focused on heading or depth control.

There has been additional work done on model-based closed-loop motion control to achieve maneuvering or trajectory tracking

[35–41]. In Ref. [35], a point-to-point control of a four-link robotic fish was implemented, where a classical proportional-integral derivative controller and a fuzzy logic controller were designed for speed and orientation control. The authors in Ref. [36] devised a control strategy for maneuvering an aquatic vehicle using an oscillating foil. The strategy consists of an optimal off-line motion planning step and an online feedback control step composed of a cascade of finite time, time-scalable linear quadratic control and input–output linearization, in combination with a sliding mode controller. Furthermore, in Ref. [37] a target-tracking and collision-avoidance algorithm for two autonomous robotic fish was implemented via a situated-behavior-based decentralized control approach, using a combination of an attractive force toward a target and a repulsive force for collision avoidance. In Ref. [38], a fuzzy control law for a pectoral-fin-driven robotic fish was developed to perform rendezvous and docking with an underwater post in water currents. Zou et al. developed a neural-network-based sliding mode control algorithm for cooperative trajectory tracking of multiple robotic fish [39]. In Ref. [40], three simplified linearized models of the decoupled fish dynamics were used for the design of linear quadratic regulators to achieve

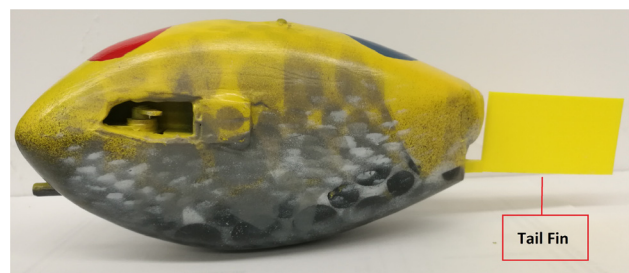


Fig. 1 A robotic fish developed by the Smart Microsystems Lab at Michigan State University

¹Corresponding author.

Contributed by the Dynamic Systems Division of ASME for publication in the JOURNAL OF DYNAMIC SYSTEMS, MEASUREMENT, AND CONTROL. Manuscript received May 5, 2018; final manuscript received March 8, 2019; published online April 9, 2019. Assoc. Editor: Vladimir Vantsevich.

speed and orientation control and to stabilize the pitch and roll. Furthermore, a line-of-sight guidance scheme was implemented for way-point tracking. Finally, the authors in Ref. [41] designed a sliding mode controller for swimming, orientating, and way-point tracking of robotic fish in three-dimensional motion. Despite aforementioned progress in the control of robotic fish, a unified, systematic control approach that incorporates performance objectives and accommodates input constraints for such robots has not been proposed.

Nonlinear model predictive control (NMPC) presents a promising framework for dealing with uncertainties as well as input and state constraints. There is extensive work on NMPC for path-following of mobile robots [42–47], but little work has been reported on its application to control of robotic fish. In this work, we propose and implement a path-following NMPC scheme for a tail-actuated robotic fish. We consider a tail-actuated robotic fish particularly because of its simple mechanical design and low power consumption. On the other hand, the highly nonlinear and coupled dynamics, along with the under-actuated nature of the robot, pose significant challenges for the control design.

In this controller design, a high-fidelity averaged nonlinear dynamic model is used. Furthermore, the physical control inputs consist of two of the tail-beat parameters, the bias and the amplitude, while the other (angular frequency) is kept constant. We propose a framework to address the nonlinear input constraints. Specifically, to maximize the use of the admissible control and handle the nonlinear control constraints in a computationally efficient manner, we employ an analytical projection scheme for the control inputs. We further propose a novel estimation scheme to identify some key unknown parameters in the robotic fish model. In particular, inspired by the work in Ref. [48], we develop a parameter estimation method to empirically identify the hydrodynamic and scaling coefficients of the model instead of utilizing time-consuming computational fluid dynamics simulations, or relying on trial-and-error data fitting between dynamic simulation and experimental measurement. To implement the controller in real-time, we employ a framework using Visual C++, which consists of the ACADO toolkit [49] used to solve repeatedly the optimal control problem, and an image processing algorithm using OpenCV to provide feedback.

Some preliminary results of this work were presented at the 2016 ASME Dynamic Systems and Control Conference [50]. The improvement of this paper over [50] is extensive and significant. First, we have proposed a computation-efficient method to deal with the nonlinear constraints. Second, we have further formulated a parameter estimation scheme to identify crucial parameters in the model. Third, we have developed the experimental framework and implemented the proposed NMPC scheme experimentally.

The rest of the paper is organized as follows. We first review the dynamic and scaled averaging models of the tail-actuated robotic fish, followed by a simplified averaged model. In Sec. 3, we present the path-following problem formulation, followed by the NMPC design and the proposed control projection scheme. In Sec. 4, simulation results are discussed, and in Sec. 5, the experimental setup, the proposed parameter estimation scheme, and the experimental results are presented. Section 6 concludes the paper.

2 Robotic Fish Model

2.1 Dynamic Model. As was done in Ref. [48], the tail-actuated robotic fish is modeled as a rigid body with a rigid tail that is actuated at its base, and it is assumed that the robot operates in an inviscid, irrotational, and incompressible fluid within an infinite domain. Define $[X, Y, Z]^T$ and $[x, y, z]^T$ as the inertial coordinate system and the body-fixed coordinate system, respectively, as illustrated in Fig. 2. The velocity of the center of mass in the body-fixed coordinates is expressed as $\mathbf{V}_c = [V_{c_x}, V_{c_y}, V_{c_z}]^T$, where

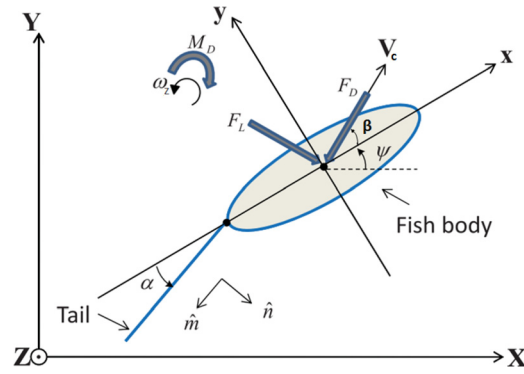


Fig. 2 Top view of the tail-actuated robotic fish undergoing planar motion [48]

V_{c_x} , V_{c_y} , and V_{c_z} denote the surge, sway, and heave velocities, respectively. Furthermore, let β denote the angle of attack, formed by the direction of $\mathbf{V}_c$ with respect to the x -axis, and ψ denote the heading angle, formed by the x -axis relative to the X -axis. The angular velocity expressed in the body-fixed coordinate system is given by $\omega = [\omega_x, \omega_y, \omega_z]^T$, which is composed of roll (ω_x), pitch (ω_y), and yaw (ω_z). Finally, let α denote the tail deflection angle with respect to the negative x -axis.

We only consider the planar motion and further assume that the body is symmetric with respect to the xz -plane and that the tail moves in the xy -plane. As a result, the system only has three degrees-of-freedom, surge (V_{c_x}), sway (V_{c_y}), and yaw (ω_z). It is further assumed that the inertial coupling between yaw, sway and surge motions is negligible, which leads to the following equations of planar motion:

$$(m_b - m_{a_x})\dot{V}_{c_x} = (m_b - m_{a_x})V_{c_y}\omega_z + f_x \quad (1)$$

$$(m_b - m_{a_y})\dot{V}_{c_y} = -(m_b - m_{a_x})V_{c_x}\omega_z + f_y \quad (2)$$

$$(J_{b_z} - J_{a_z})\dot{\omega}_z = (m_{a_y} - m_{a_x})V_{c_x}V_{c_y} + M_z \quad (3)$$

where m_b is the mass of the body, J_{b_z} is the inertia of the body about the z -axis, m_{a_x} and m_{a_y} are the hydrodynamic derivatives that represent the added masses of the robotic fish along the x and y directions, respectively, and J_{a_z} represents the added inertia effect of the body about the z -axis. The hydrodynamic forces and moment due to tail fin actuation and the interaction of the body itself with the fluid are captured by f_x , f_y , and M_z . To evaluate the hydrodynamic forces exerted by the tail, Lighthill's large amplitude elongated body theory is used [48]. The kinematic equations for the robotic fish are given by

$$\dot{X} = V_{c_x} \cos \psi - V_{c_y} \sin \psi \quad (4)$$

$$\dot{Y} = V_{c_x} \sin \psi + V_{c_y} \cos \psi \quad (5)$$

$$\dot{\psi} = \omega_z \quad (6)$$

Given the rhythmic nature of the robotic fish movement and the periodic tail actuation, averaging has proven to be a useful approach in studying the effect of the input parameters on the dynamics of the robotic fish [48]. Furthermore, in practical applications, it is more natural to control the parameters for periodic fin beats than to directly control the fin position at every moment. Therefore, an averaged model is best suited for trajectory planning and tracking control. We next review the averaged model proposed in Ref. [48], where the following periodic pattern for the tail deflection angle is considered:

$$\alpha(t) = \alpha_0 + \alpha_a \sin(\omega_\alpha t) \quad (7)$$

where α_0 , α_a , and ω_α represent the bias, amplitude, and frequency of the tail beat, respectively. The original hydrodynamic force and moment terms in Eqs. (1)–(3) are scaled by some functions dependent on the tail beat parameters, α_0 , α_a , and ω_α , and classical averaging is then conducted over these scaled dynamics. In particular, we define the states $x_1 = V_{c_x}$, $x_2 = V_{c_y}$, and $x_3 = \omega_z$, so that the averaged dynamics takes the following form:

$$\dot{x}_1 = f_1(x_1, x_2, x_3) + K_f \bar{f}_4(\alpha_0, \alpha_a, \omega_\alpha) \quad (8)$$

$$\dot{x}_2 = f_2(x_1, x_2, x_3) + K_f \bar{f}_5(\alpha_0, \alpha_a, \omega_\alpha) \quad (9)$$

$$\dot{x}_3 = f_3(x_1, x_2, x_3) + K_m \bar{f}_6(\alpha_0, \alpha_a, \omega_\alpha) \quad (10)$$

with

$$f_1(x_1, x_2, x_3) = \frac{m_2}{m_1} x_2 x_3 - \frac{c_1}{m_1} x_1 \sqrt{x_1^2 + x_2^2} + \frac{c_2}{m_1} x_2 \sqrt{x_1^2 + x_2^2} \arctan\left(\frac{x_2}{x_1}\right) \quad (11)$$

$$f_2(x_1, x_2, x_3) = -\frac{m_1}{m_2} x_1 x_3 - \frac{c_1}{m_2} x_2 \sqrt{x_1^2 + x_2^2} - \frac{c_2}{m_2} x_1 \sqrt{x_1^2 + x_2^2} \arctan\left(\frac{x_2}{x_1}\right) \quad (12)$$

$$f_3(x_1, x_2, x_3) = (m_1 - m_2) x_1 x_2 - c_4 \omega_z^2 \operatorname{sgn}(\omega_z) \quad (13)$$

$$\bar{f}_4(\alpha_0, \alpha_a, \omega_\alpha) = \frac{mL^2}{12m_1} \omega_\alpha^2 \alpha_a \left(3 - \frac{3}{2} \alpha_0^2 - \frac{3}{8} \alpha_a^2\right) \quad (14)$$

$$\bar{f}_5(\alpha_0, \alpha_a, \omega_\alpha) = \frac{mL^2}{4m_2} \omega_\alpha^2 \alpha_a^2 \alpha_0 \quad (15)$$

$$\bar{f}_6(\alpha_0, \alpha_a, \omega_\alpha) = -\frac{cmL^2}{4J_3} \omega_\alpha^2 \alpha_a^2 \alpha_0 \quad (16)$$

where $m_1 = m_b - m_a$, $m_2 = m_b - m_{a_y}$, $J_3 = J_{b_z} - J_{a_z}$, $c_1 = (1/2)\rho S C_D$, $c_2 = (1/2)\rho S C_L$, $c_3 = (1/2)mL^2$, $c_4 = (1/(J_3))K_D$, $c_5 = (1/(2J_3))L^2 mc$, and $c_6 = (1/(3J_3))L^3 m$. Here S denotes the reference surface area for the robot body, C_D , C_L , and K_D represent the drag force coefficient, lift coefficient, and drag moment coefficient, respectively, ρ is the density of water, L is the tail length, c is the distance from the body center to the pivot point of the actuated tail, and m represents the mass of water displaced by the tail per unit length and is approximated by $(\pi/4)\rho d^2$ with d denoting the tail depth. K_f is a scaling constant, and $K_m(\alpha_0)$ is a scaling function affine in α_0 . To further facilitate control design, in this paper K_m is considered as a constant during the NMPC design. This term is found by taking the average of K_m for a given range of α_0 . The resulting model is called the simplified averaged model in this paper.

3 Path-Following Control Algorithm

Considering that robotic fish are battery-powered, energy-efficient locomotion is highly desirable in order to prolong the field-operation time. It is important to design a controller that is able to meet performance objectives such as minimizing the path-tracking error while accommodating consideration of control effort. We are thus motivated to develop an NMPC scheme for path-following. NMPC is an attractive choice because it allows explicit consideration of state and input constraints, is capable of handling nonlinear models, and can optimize control performance [51,52].

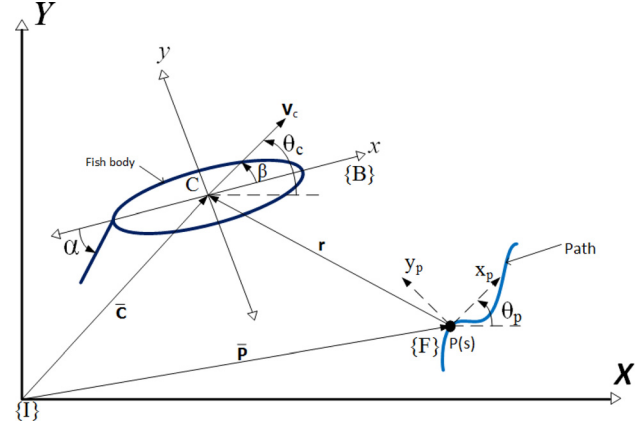


Fig. 3 Illustration of the path-following problem for robotic fish

3.1 Path-Following Error Coordinates. In contrast to trajectory tracking, in path-following one is interested in following a geometric reference parametrized by some scalar without any specified timing. The kinematic model of the robotic fish is expressed in a Frenet-Serret frame $\{F\}$ that moves along the reference path according to some desired function of time. Figure 3 illustrates the path-following problem.

Assume that the reference path is a twice continuously differentiable geometric curve that is defined as a set of points P parametrized by the scalar s

$$P = \{P \in \mathbb{R}^2 | P = p(s), \forall s \in [0, l_p]\} \quad (17)$$

where l_p denotes the length of the path, and the function $p: \mathbb{R}^1 \rightarrow \mathbb{R}^2$ is twice differentiable. Let P denote a point on the path to be followed, θ_p the tangential angle of the path at point P , and θ_c the angle between the robotic fish velocity vector V_c and the inertial X -axis, while the coordinate axes x_p and y_p are directed along the tangential and normal directions at point P . We let point C denote the center of the robotic fish, and the vectors $\bar{C}$ and $\bar{P}$ describe the positions of C and P in the three-dimensional (3D) inertial frame $\{I\}$. Note that since we are only considering the planar case, the third component of the position vectors is taken as 0. Let $r = [x_e; y_e; 0]^T$ denote the position of the robotic fish center C with respect to the point P on the path expressed in $\{F\}$. Let ${}^I R_F$ denote the rotation matrix from $\{I\}$ to $\{F\}$ and ${}^F R_I$ denote the rotation matrix from $\{F\}$ to $\{I\}$, with

$${}^I R_F = \begin{bmatrix} \cos \theta_p & \sin \theta_p & 0 \\ -\sin \theta_p & \cos \theta_p & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (18)$$

Define $\dot{\theta}_p = C_p(s)\dot{s}$, where $C_p(s)$ is the path's curvature. One can express

$$\dot{\bar{C}} = \dot{\bar{P}} + {}^F R_I \dot{r} \quad (19)$$

The velocity of $\bar{C}$ in $\{I\}$ is given by

$$\left(\frac{d\bar{C}}{dt}\right)_{\{I\}} = \left(\frac{d\bar{P}}{dt}\right)_{\{I\}} + {}^F R_I \left(\frac{dr}{dt}\right)_{\{F\}} + {}^F R_I (\omega_p \times r) \quad (20)$$

where

$$\omega_p = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_p \end{bmatrix} \quad (21)$$

Multiplying the previous equation on the left by ${}^I\mathbf{R}_F$ gives the velocity of C expressed in {F}

$${}^I\mathbf{R}_F \left(\frac{d\bar{\mathbf{C}}}{dt} \right)_{\{I\}} = \left(\frac{d\bar{\mathbf{P}}}{dt} \right)_{\{F\}} + \left(\frac{d\mathbf{r}}{dt} \right)_{\{F\}} + (\boldsymbol{\omega}_p \times \mathbf{r}) \quad (22)$$

where

$$\left(\frac{d\bar{\mathbf{C}}}{dt} \right)_{\{I\}} = \begin{bmatrix} \dot{X} \\ \dot{Y} \\ 0 \end{bmatrix} \quad (23)$$

$$\left(\frac{d\bar{\mathbf{P}}}{dt} \right)_{\{F\}} = \begin{bmatrix} \dot{s} \\ 0 \\ 0 \end{bmatrix} \quad (24)$$

$$\left(\frac{d\mathbf{r}}{dt} \right)_{\{F\}} = \begin{bmatrix} \dot{X}_e \\ \dot{Y}_e \\ 0 \end{bmatrix} \quad (25)$$

$$\boldsymbol{\omega}_p \times \mathbf{r} = \begin{bmatrix} 0 \\ 0 \\ C_p(s)\dot{s} \end{bmatrix} \times \begin{bmatrix} X_e \\ Y_e \\ 0 \end{bmatrix} = \begin{bmatrix} -C_p(s)\dot{s}Y_e \\ C_p(s)\dot{s}X_e \\ 0 \end{bmatrix} \quad (26)$$

After rearranging and solving for $\dot{X}_e$ and $\dot{Y}_e$ from Eq. (22), we get the following expression:

$$\begin{bmatrix} \dot{X}_e \\ \dot{Y}_e \end{bmatrix} = \begin{bmatrix} \dot{X} \cos \theta_p + \dot{Y} \sin \theta_p - \dot{s} + C_p(s)\dot{s}Y_e \\ -\dot{X} \sin \theta_p + \dot{Y} \cos \theta_p - C_p(s)\dot{s}X_e \end{bmatrix} \quad (27)$$

Let $\alpha_e = \psi - \theta_p$, and further expand $\dot{X}$ and $\dot{Y}$ with Eqs. (4) and (5), respectively, which results in the following error state model:

$$\begin{pmatrix} \dot{X}_e \\ \dot{Y}_e \\ \dot{\alpha}_e \end{pmatrix} = \begin{pmatrix} V_c \cos(\alpha_e + \beta) - \dot{s} + C_p(s)\dot{s}Y_e \\ V_c \sin(\alpha_e + \beta) - C_p(s)\dot{s}X_e \\ \omega_z - C_p(s)\dot{s} \end{pmatrix} \quad (28)$$

where V_c is the magnitude of the robotic fish's translational velocity. The dynamical model of the robotic fish in the error state is then obtained by augmenting the previous equations with the simplified averaged scaled dynamics as seen in Eqs. (8)–(10). Ideally, one would like the robotic fish to not only converge to a desired path, but also move along the path with some desired surge velocity and desired angular velocity. Let V_{d_s} be the desired surge velocity, and let the velocity error states be defined by $\eta_e = V_{c_x} - V_{d_s}$, $\omega_e = \omega_z - C_p(s)\dot{s}$, and $\zeta_e = \dot{s} - V_c \cos(\alpha_e + \beta)$ so that

$$\begin{pmatrix} \dot{\eta}_e \\ \dot{\omega}_e \\ \dot{\zeta}_e \end{pmatrix} = \begin{pmatrix} \dot{V}_{c_x} - \dot{V}_{d_s} \\ \dot{\omega}_z - C_p(s)\dot{s} - g_c(s)\dot{s}^2 \\ \dot{s} - \dot{V}_c \cos(\alpha_e + \beta) + V_c \sin(\alpha_e + \beta)(\dot{\alpha}_e + \dot{\beta}) \end{pmatrix} \quad (29)$$

where $g_c(s) = ((dC_p(s))/ds)$. Since we are interested in steering the robotic fish such that $V_{c_x} = V_{d_s}$, and $\omega_z = C_p(s)\dot{s}$, by doing a change of variables on Eqs. (8)–(10) using the previous definitions, we can express the robotic fish dynamic equations in terms of the error velocity states. The error state vector is then given by

$$\boldsymbol{\Omega}_e = \begin{pmatrix} X_e \\ Y_e \\ \alpha_e \\ \eta_e \\ \omega_e \\ \zeta_e \end{pmatrix} \quad (30)$$

Since we have formulated the problem with respect to the error dynamics, and have shifted the equilibrium point of the dynamic equations, our control objective has become a stabilization problem for the resultant error dynamics.

3.2 Path-Following Control Design. To steer the robotic fish to the desired path, and drive the error state vector $\boldsymbol{\Omega}_e$ to zero, we utilize an NMPC scheme using the robot's simplified averaged model. NMPC is an optimization-based method for feedback control of nonlinear systems, where the basic idea is to repeatedly solve a finite horizon optimization problem subject to state and input constraints. At a given time t , measurements are obtained, and using a model of the process, the controller predicts the behavior of the system over a prediction horizon T_p and then determines over the control horizon T_c the input necessary to maximize the performance objective. The first part of the optimal control obtained is implemented until the next sampling instant, and then a new measurement is obtained and the process repeats [52].

To design the controller, we consider the robot's simplified averaged model in which the control represents functions of the actual control variables, namely, the tail-beat pattern parameters α_0 , α_a , and ω_x . By choosing the control in this manner, we allow the control inputs to appear linearly in the dynamic equation. In particular, we have chosen our control inputs as

$$u_{f_1} = \alpha_a^2 \left(3 - \frac{3}{2}\alpha_0^2 - \frac{3}{8}\alpha_a^2 \right) \quad (31)$$

$$u_{f_2} = \alpha_a^2 \alpha_0 \quad (32)$$

which are present in functions $f_4(\alpha_0, \alpha_a, \omega_x)$ to $f_6(\alpha_0, \alpha_a, \omega_x)$ in Eqs. (8)–(10). To simplify discussion, it is assumed that the robotic fish uses a fixed tail-beat frequency ω_x . Furthermore, since the system dynamics are expressed in terms of the velocity errors states, by doing a change of variables we have essentially shifted the equilibrium point of the dynamical system, which means that there is also a shift on the control inputs u_{f_1} and u_{f_2} . Let u_{2ss} and u_{3ss} represent the shifted control values, which are defined as follows:

$$u_{2ss} = \frac{6\rho s C_D V_{d_s}^2}{K_f m L^2 \omega_x^2} \quad (33)$$

$$u_{3ss} = \frac{4K_D C_P(s) V_{d_s}^2}{K_m m L^2 c \omega_x^2} \quad (34)$$

In order to satisfy the condition that $\dot{\boldsymbol{\Omega}}_e = 0$ when $\boldsymbol{\Omega}_e = 0$ and $\mathbf{u}_e = 0$, we define the control inputs as follows:

$$\mathbf{u}_e = \begin{pmatrix} u_{e1} \\ u_{e2} \\ u_{e3} \end{pmatrix} = \begin{pmatrix} \dot{s} - \dot{V}_c \cos(\alpha_e + \beta) + V_c \sin(\alpha_e + \beta)(\dot{\alpha}_e + \dot{\beta}) \\ u_{f_1} - u_{2ss} \\ u_{f_2} - u_{3ss} \end{pmatrix} \quad (35)$$

where u_{e1} is essentially $\dot{\zeta}_e$ as seen in Eq. (29).

Since one is interested in steering the robotic fish to the desired path, we employ a stage cost that is a function of the error state vector $\boldsymbol{\Omega}_e$. Furthermore, to minimize the control effort a weighting term on the control inputs is introduced. The following quadratic cost is chosen:

$$F(\boldsymbol{\Omega}_e, \mathbf{u}_e) = (\boldsymbol{\Omega}_e)^T \mathbf{Q}(\boldsymbol{\Omega}_e) + (\mathbf{u}_e)^T \mathbf{R}(\mathbf{u}_e) \quad (36)$$

where Q and R are positive definite weighting matrices that penalize deviations from the desired values.

Furthermore, to guarantee closed-loop stability and convergence, we utilize the terminal penalty, and the fictitious terminal control law $\pi(\Omega_e)$ as proposed in Ref. [53], where a polytopic linear differential inclusion-based method is employed to obtain the weighting matrix Q_T for a terminal penalty of the following form:

$$E(\Omega_e(t + T_p)) = (\Omega_e(t + T_p))^T Q_T (\Omega_e(t + T_p)) \quad (37)$$

The reader is referred to Ref. [53] for details on how to obtain this weighting matrix.

By solving the optimal control problem we obtain the optimal control sequence for u_{e1} , u_{e2} , and u_{e3} . From u_{e1} , we can obtain ζ_e , and thus the state ζ_e from which we can then solve for $\dot{s}$. Furthermore, from u_{e2} and u_{e3} , along with Eqs. (31)–(34), one can solve for the actual robotic fish control variables α_0 and α_a .

3.3 Control Projection. Given that the NMPC inputs, u_{e2} and u_{e3} , consist of functions of the actual robotic fish control variables α_0 and α_a , the NMPC input constraints are nonlinear in nature. As an illustration, Fig. 4 plots the admissible control inputs in terms of u_{f1} and u_{f2} when the tail beat bias and amplitude have the following limits:

- $\alpha_{0\min} = -40$ deg
- $\alpha_{0\max} = 40$ deg
- $\alpha_{a\min} = 0$ deg
- $\alpha_{a\max} = 30$ deg

where $\alpha_{0\min}$, $\alpha_{0\max}$, $\alpha_{a\min}$, and $\alpha_{a\max}$ are the physical limits on the tail-beat bias and amplitude, respectively.

Although NMPC is able to handle nonlinear constraints, defining the constraints in this manner leads to an increase in computational time and complexity which in turn makes it challenging to implement in real-time. It is thus desirable to define boxed-constraints since this can reduce significantly the complexity of the optimization problem and thus lower the computational time. One way of handling the irregular sector-shaped admissible control region as shown in Fig. 4, is to choose a rectangular area that lies inside this sector-shaped region as depicted by the light gray box in Fig. 5. However, this deprives one of fully utilizing the admissible control. To overcome this problem, we propose to employ a projection method, where we define the NMPC boxed constraint to be such that it encompasses the admissible control region as depicted in Fig. 6, and then project the computed values onto the true region depicted by the red sector-shaped section.

Let z denote a control point anywhere in this rectangular region, and let the sector-shaped set be represented by U . We can then project the point z onto the convex set U such that

$$\text{Proj}_U(z) \triangleq \arg \min_{u \in U} \|z - u\| \quad (38)$$

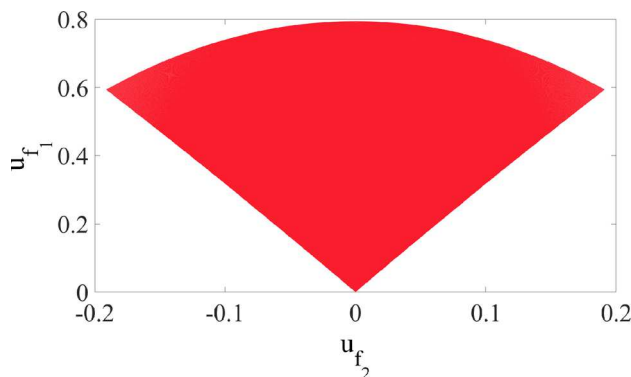


Fig. 4 Illustration of the admissible input control set for robotic fish

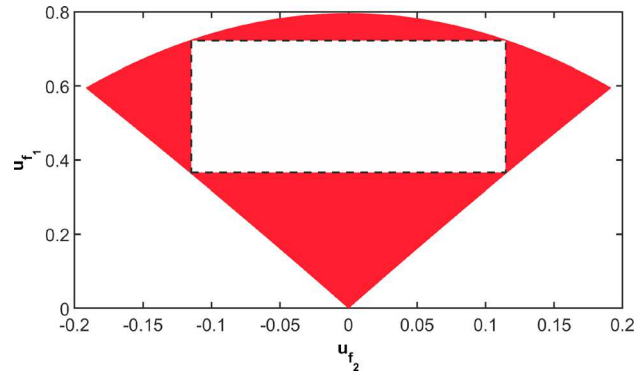


Fig. 5 Illustration of a rectangular section chosen from the admissible control set

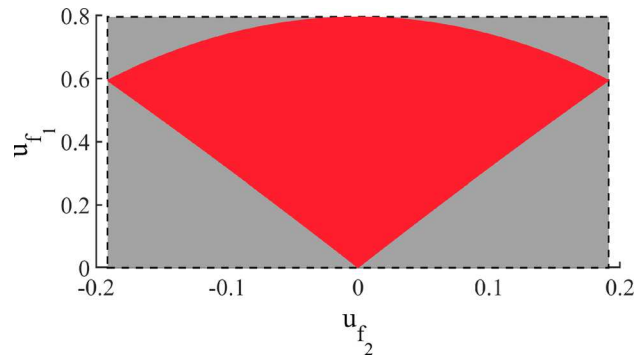


Fig. 6 Boxed section encompassing the admissible control set

Given that U is convex, this problem is then well defined and $\text{Proj}_U(z)$ is unique. Instead of relying on an iterative optimization algorithm to determine the projected value, one can directly obtain an analytical solution that will simplify the projection and minimize computational complexity, as explained next. By taking advantage of the symmetry of the admissible control set, one can restrict the analysis to the left-half plane. To characterize the boundaries of the admissible control region, we obtain the relationship between u_{f1} , u_{f2} , and α_0 by solving for α_a from Eq. (32) and then substituting that into Eq. (31). Similarly, by solving for α_0 from Eq. (32), we can obtain an equation that captures the relationship between u_{f1} , u_{f2} , and α_a . These equations are given as follows:

$$\chi_1(u_{f1}, u_{f2}, \alpha_0) \triangleq -\alpha_0^2 u_{f1} + \left(3 - \frac{3}{2} \alpha_0^2\right) \alpha_0 u_{f2} - \frac{3}{8} u_{f2}^2 = 0 \quad (39)$$

$$\chi_2(u_{f1}, u_{f2}, \alpha_a) \triangleq -\alpha_a^2 u_{f1} - \frac{3}{2} u_{f2}^2 + 3\alpha_a^4 - \frac{3}{8} \alpha_a^6 = 0 \quad (40)$$

where Eq. (39) represents the left boundary when $\alpha_0 = \alpha_{0\min}$ and Eq. (40) represents the arc at the top when $\alpha_a = \alpha_{a\max}$. To implement the projection scheme, the following cases are considered:

$$\left\{ \begin{array}{l} (A) \chi_1(u_{f1}, u_{f2}, \alpha_{0\min}) \leq 0 \text{ and } \chi_2(u_{f1}, u_{f2}, \alpha_{a\max}) \leq 0 \\ (B) \chi_1(u_{f1}, u_{f2}, \alpha_{0\min}) \leq 0 \text{ and } \chi_2(u_{f1}, u_{f2}, \alpha_{a\max}) > 0 \\ (C) \chi_1(u_{f1}, u_{f2}, \alpha_{0\min}) > 0 \text{ and } \chi_2(u_{f1}, u_{f2}, \alpha_{a\max}) \leq 0 \\ (D) \chi_1(u_{f1}, u_{f2}, \alpha_{0\min}) > 0 \text{ and } \chi_2(u_{f1}, u_{f2}, \alpha_{a\max}) > 0 \end{array} \right.$$

For case (A), the point to be projected is inside or on the boundary of the convex set U and no projection is needed. For case (D), the point to be projected would be outside of the box encompassing the constraint set U and thus does not need to be considered. For case (B), the point to be projected is above the arc, and therefore $\text{Proj}_U(z)$ can be found by finding the minimum distance from the point z to the arc described by Eq. (40). Let $z = (p, q)$ and $u^* = (u_{f_1}^*, u_{f_2}^*) = \text{Proj}_U(z)$. The relationship between $u_{f_1}^*$ and $u_{f_2}^*$ is given by

$$u_{f_1}^* = \frac{1}{\alpha_{d_{\max}}^2} \left(\frac{-3}{2} u_{f_2}^{*2} + 3\alpha_{d_{\max}}^4 - \frac{3}{8} \alpha_{d_{\max}}^6 \right) \quad (41)$$

and the distance between z and u^* is then given by

$$\begin{aligned} g_2(u_{f_2}^*) &= (u_{f_2}^* - p)^2 + (u_{f_1}^* - q)^2 \\ &= (u_{f_2}^* - p)^2 \\ &\quad + \left(\frac{1}{\alpha_{d_{\max}}^2} \left(\frac{-3}{2} u_{f_2}^{*2} + 3\alpha_{d_{\max}}^4 - \frac{3}{8} \alpha_{d_{\max}}^6 \right) - q \right)^2 \end{aligned} \quad (42)$$

By taking the partial derivative of $g_2(u_{f_2}^*)$ with respect to $u_{f_2}^*$ and setting it to zero, we can obtain a unique real root for $u_{f_2}^*$ that would minimize this distance. Finally, $u_{f_1}^*$ is obtained with Eq. (41).

For case (C), the point to be projected is below the left boundary. In this case, $u_{f_1}^*$ and $u_{f_2}^*$ are related by

$$u_{f_1}^* = \frac{1}{\alpha_{0_{\min}}^2} \left(\frac{-3\alpha_{0_{\min}}^3}{2} u_{f_2}^* + 3\alpha_{0_{\min}} u_{f_2}^* - \frac{3}{8} u_{f_2}^{*2} \right) \quad (43)$$

and the distance between z and u^* can be captured by

$$\begin{aligned} g_3(u_{f_2}^*) &= (u_{f_2}^* - p)^2 + (u_{f_1}^* - q)^2 \\ &= (u_{f_2}^* - p)^2 + \left(\frac{-3\alpha_{0_{\min}}}{2} u_{f_2}^* + \frac{3u_{f_2}^*}{\alpha_{0_{\min}}^2} - \frac{3}{8\alpha_{0_{\min}}^2} u_{f_2}^{*2} - q \right)^2 \end{aligned} \quad (44)$$

By taking the partial derivative of $g_3(u_{f_2}^*)$ with respect to $u_{f_2}^*$ and setting it to zero, we can obtain a unique real root for $u_{f_2}^*$, and consequently $u_{f_1}^*$ with Eq. (43).

4 Simulation Results

To evaluate the effectiveness of the designed controller, simulation was carried out using ACADO model predictive control toolkit. The parameters used (Table 1) were based on a robotic fish developed by Smart Microsystems Lab at Michigan State University. Furthermore, while the input constraints are the same as those presented in the experiment section, the parameters used to solve the optimization problem and implement the NMPC are as follows:

- Length of optimization horizon : $T_c = T_p = 12$ s
- Sampling interval : $t_s = 1$ s
- Weighting matrix : $Q = \text{diag}(7, 7, 0.3, 1, 1, 7)$
- Control weighting matrix : $R = 0.001I_3$
- $V_{c \max} = 0.04$ m/sec
- $\dot{s}_{\max} = 0.04$ m/sec
- $\alpha_{0_{\min}} = -40$ deg
- $\alpha_{0_{\max}} = 40$ deg
- $\alpha_{a_{\min}} = 0$ deg
- $\alpha_{a_{\max}} = 30$ deg

where $V_{c \max}$ is the maximum velocity the robotic fish can achieve, $\dot{s}_{\max}$ is the maximum speed the point s can move along the path with, and $\alpha_{0_{\min}}$, $\alpha_{0_{\max}}$, $\alpha_{a_{\min}}$, and $\alpha_{a_{\max}}$ are the physical

Table 1 Identified parameters for the robotic fish used in this work

Parameter	Value
m_b	0.725 kg
m_{ax}	-0.217 kg
m_{ay}	-0.7888 kg
J_{bz}	2.66×10^{-3} kg/m ²
J_{az}	-7.93×10^{-4} kg/m ²
L	0.071 m
d	0.04 m
c	0.105 m
ρ	1000 kg/m ³
S	0.03 m ²
C_D	0.97
C_L	3.9047
K_D	4.5×10^{-3} kg/m ²
K_f	0.7
K_m (averaged)	0.45

limits on the tail-beat bias and amplitude, respectively. Note that all of the following simulation was run with the same set of parameters and initial conditions. Furthermore, the terminal penalty weighting matrix was determined as described in Sec. 3.2. Though the controller was designed using the simplified averaged model, the simulation was performed using the original dynamic model. In other words, the model of the process was based on the simplified averaged dynamics as described by Eqs. (8)–(10), and the inputs obtained from solving the optimization problem were applied to the system described by Eqs. (1)–(3).

We first considered the following path:

$$\begin{aligned} x_p &= s \\ y_p &= 0 \end{aligned} \quad (45)$$

where x_p and y_p represent the position of the point P in the $\{I\}$ frame. This path has a curvature of $c_p(s) = 0$, and we chose to require the robotic fish to move with a constant velocity $V_c = 0.03$ m/s. In Figs. 7–9 we compare the desired path and the closed-loop trajectory of the robotic fish for three cases. In particular, Fig. 7 shows the simulation results of the NMPC utilizing the projection scheme, while Fig. 8 shows the results for the case when no projection was employed and a boxed constraint within U was chosen instead (as shown in Fig. 5). Finally, Fig. 9 shows the results when the nonlinear constraints for the set U were directly defined. Note that in this work the blue dashed line represents the closed-loop trajectory of the robotic fish while the solid red line represents the desired path, and the arrowheads point in the direction of progression. Furthermore, the red diamond represents the starting position of the robotic fish, the green dot represents the starting point of the path, and the magenta box represents the imaginary boundaries of the fish tank. Moreover, Fig. 10 shows the computed physical inputs from solving the

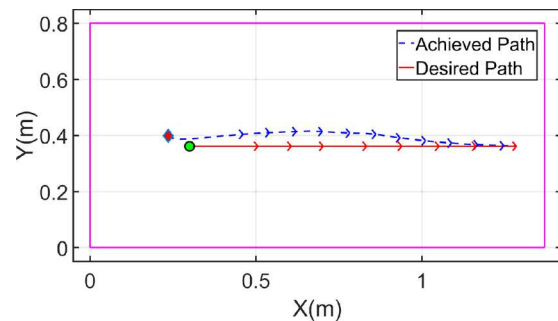


Fig. 7 Simulation: line-tracking results for NMPC with the proposed control projection method

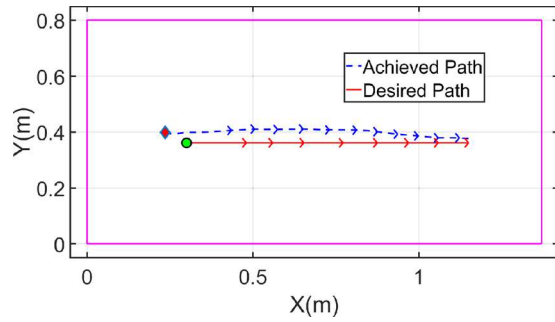


Fig. 8 Simulation: line-tracking results for NMPC with boxed constraints inside the admissible sector-shaped control region

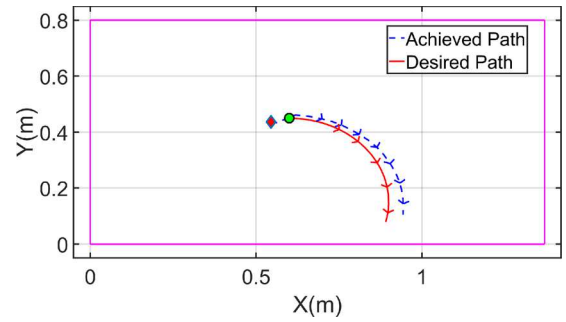


Fig. 12 Simulation: arc-tracking results for NMPC with boxed constraints inside the admissible sector-shaped control region

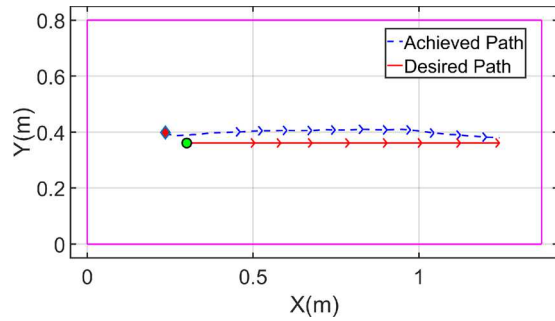


Fig. 9 Simulation: line-tracking results for NMPC with direct nonlinear control constraints

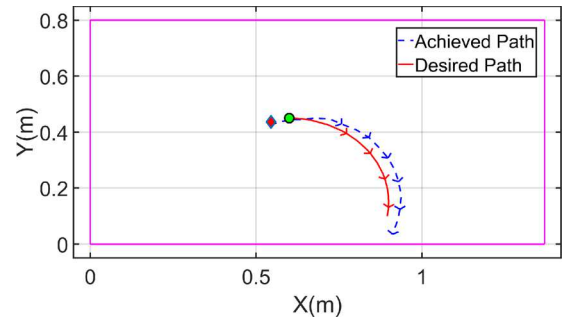


Fig. 13 Simulation: arc-tracking results for NMPC with nonlinear control constraints

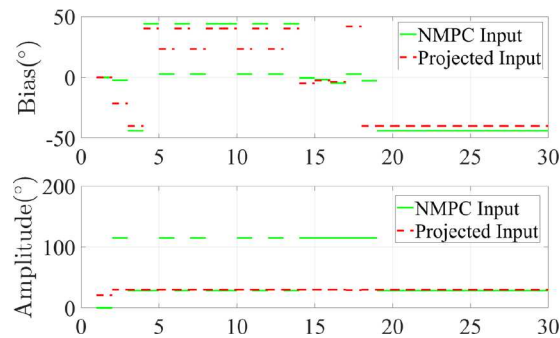


Fig. 10 Simulation: computed NMPC physical inputs and their projected values

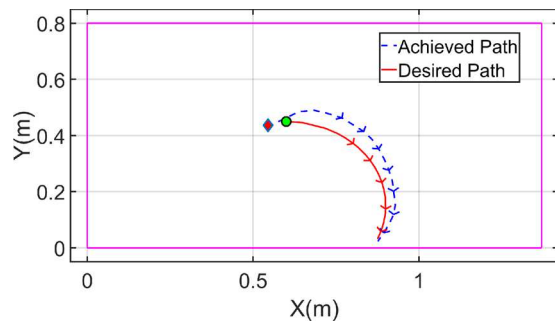


Fig. 11 Simulation: arc-tracking results for NMPC with the proposed control projection method

NMPC with the larger boxed constraint and their final values after the proposed projection.

Similarly, we considered the following circular path:

$$\begin{aligned} x_p &= 0.3 \sin(s) \\ y_p &= 0.3 \cos(s) \end{aligned} \quad (46)$$

which has a constant curvature of $c_p(s) = 3.33$. In Figs. 11–13, we compare the desired path trajectory with those obtained by the robotic fish using the three aforementioned control schemes, respectively.

From the simulation results, one can see that the proposed NMPC scheme with projection outperforms the other two schemes in both line-tracking and arc-tracking cases; in particular, it results in smaller tracking error at the steady-state. Compared with the case with boxed constraint within the set U , the proposed scheme offers larger control authority. The better tracking results from the proposed scheme compared to the case using direct, non-linear constraints, however, were somewhat surprising. We conjecture that this is because the latter algorithm cannot reach an optimal solution within the allotted computing time. In particular, directly defining the nonlinear constraints requires the optimization algorithm to conduct more iterations in order to find the solution, which also makes it difficult to implement in real-time.

5 Experimental Results

In order to evaluate the effectiveness of the designed controller, experiments were carried out using the robotic fish depicted in Fig. 1. The robot consisted of a rigid-shell body and a relatively rigid tail, which were both 3D-printed. The tail was actuated using a Hitec digital micro waterproof servo (HS-5086WP) (Poway, CA), while a microchip digital signal processors and controller (DSPIC30F6014, Chandler, AZ) was used to control the tail actuation. Furthermore, an XBee-PRO module (Hopkins, MN) was used for communication with a computer. Two Tenergy Li-Ion rechargeable batteries (7.4V, 3350 mAh) (Fremont, CA) were used to power the robot. For the experiments, the robotic fish was run in a 1.38 m by 0.8 m tank equipped with an overhead Logitech camera (Newark, CA) as seen in Fig. 14. Furthermore, to obtain the robotic fish's position and orientation in the tank, two markers were attached to the anterior and posterior of the robotic fish body. We then captured an overhead video of the robotic fish

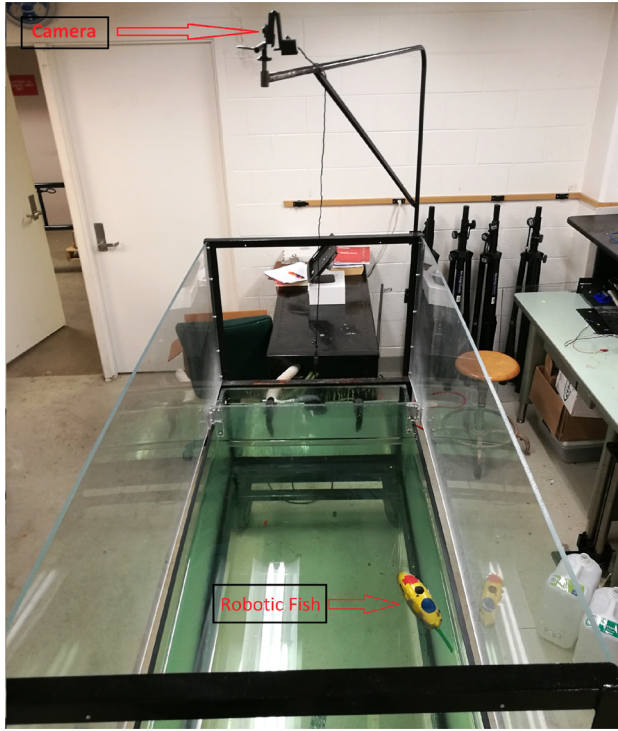


Fig. 14 The experimental setup

swimming in the tank using the camera, and utilized Visual C++ and the OpenCV library to implement an image processing algorithm. The algorithm detected the positions of the two markers and then used their average to obtain the center position of the robotic fish. The heading angle of the robot was estimated using the positions of the two markers. Additionally, the Kalman filter function in OpenCV was used to estimate the linear and angular velocities of the robot based on the measured position and heading. During every sampling time t_s , the OpenCV algorithm was used to obtain measurements for NMPC, which were then passed to the nonlinear optimization tool ACADO to solve the optimal control problem. In particular, we ran the software on a Surface Pro tablet with an Intel(R) Core(TM) i5 central processing unit @ 2.50 GHz with 4.0 GB of DDR3 RAM. Once the control inputs were calculated, the bias and amplitude values for the tail beat were obtained and then transmitted to the robotic fish wirelessly, and the process was repeated.

5.1 Model Parameter Identification. The robotic fish mass and tail fin dimensions were measured, the values of which are as shown in Table 1. Furthermore, the added masses, added inertia, and wetted surface were calculated based on a prolate spheroid approximation of the robotic fish body [54]. Identification of the hydrodynamic parameters (such as C_D , C_L , and K_D) of the robotic fish model (8)–(16) typically requires extensive effort in fitting dynamic simulation data to experimental data by scanning the parameter space [15]. Furthermore, the determination of the scaling coefficients of K_f and K_m requires scanning the parameter space for multiple sets of tail beat patterns and matching the simulated average model data to the simulated dynamic model [48], which is time-consuming. In this paper, we propose an efficient and systematic way to identify the model parameters by exploiting the approximate, analytical relationship between the steady-state turning parameter (turning radius, turning period, etc.) and the model parameters established in Ref. [48]. With the assumption $|x_1| \gg |x_2|$, which is reasonable in general, we can obtain the unique equilibrium of the system (8)–(10), under a given tail beat pattern, as

$$\bar{x}_1 = \sqrt{\frac{K_f m L^2 \omega_a^2 \alpha_a \left(3 - \frac{3}{2} \alpha_0^2 - \frac{3}{8} \alpha_a^2\right)}{6 \rho S C_D}} \quad (47)$$

$$\bar{x}_2 = \frac{K_f}{2 \rho S (C_D + C_L)} \sqrt{\frac{6 m \rho S C_D L^2 \omega_a^2 \alpha_a^2 \alpha_0^2}{K_f \left(3 - \frac{3}{2} \alpha_0^2 - \frac{3}{8} \alpha_a^2\right)}} + \frac{2 m_1}{\rho S (C_D + C_L)} \sqrt{\frac{K_m m L^2 c \omega_a^2 \alpha_a^2 \alpha_0}{4 K_D}} \quad (48)$$

$$\bar{x}_3 = -\sqrt{\frac{K_m m L^2 c \omega_a^2 \alpha_a^2 \alpha_0}{4 K_D}} \quad (49)$$

And the steady-state turning period T_p (i.e., how long the robot takes to complete a full orbit), turning radius R , and angle of attack β can be expressed as

$$T_p = 2\pi/|\bar{x}_3| \quad (50)$$

$$R = \sqrt{(\bar{x}_1^2 + \bar{x}_2^2)}/|\bar{x}_3| \quad (51)$$

$$\beta = \arctan(\bar{x}_2/\bar{x}_1) \quad (52)$$

Using Eqs. (47)–(52), we formulate the following algorithm to obtain the hydrodynamic coefficients C_D , C_L , and K_D , as well as the scaling coefficients K_m and K_f for the averaged model.

Let $R_1 = (K_f/C_D)$. By solving for the ratio (K_f/C_D) from Eq. (47), we obtain

$$R_1 = \frac{6 \rho S \bar{x}_1^2}{m L^2 \omega_a^2 \alpha_a \left(3 - \frac{3}{2} \alpha_0^2 - \frac{3}{8} \alpha_a^2\right)} \quad (53)$$

Using the previous equation, one can obtain the numerical value of the ratio R_1 for a given set of tail beat parameters and the corresponding measured $\bar{x}_1$. In particular, we found this ratio by averaging the different values obtained for each set of measurements.

Furthermore, let $K_m = K_0 + K_1 \alpha_0$. By solving for (K_m/K_D) from Eq. (49), and using the definition $\theta_0 = K_0/K_D$ and $\theta_1 = K_1/K_D$, one gets

$$\theta_0 + \alpha_0 \theta_1 = \frac{4 \bar{x}_3^2}{m L^2 c \omega_a^2 \alpha_a^2 \alpha_0} \quad (54)$$

Using Eq. (54) can then estimate the numerical values for θ_0 and θ_1 by utilizing, for example, the constrained linear least squares (lsqin) function in MATLAB, based on the tail beat parameters and the corresponding measurement of $\bar{x}_3$ for a set of experiments.

By considering R_1 and the ratio (K_m/K_D) , we have reduced the number of parameters to be estimated from 5 (K_f, K_m, C_D, C_L, K_D) to 3 ($R_1, (K_m/K_D), C_L$). In order to obtain the particular values for C_D, K_f, K_m , and K_D , and to estimate the remaining parameter C_L , we utilize Eqs. (53) and (54) along with Eqs. (48) and (51). By letting $c_0 = C_D + C_L$ and substituting R_1, θ_0 and θ_1 into Eq. (51), one obtains

$$R^2 |\bar{x}_3|^2 = \bar{x}_1^2 + \left(\frac{C_D}{c_0} d_1 + \frac{d_2}{c_0}\right)^2 \quad (55)$$

where

$$d_1 = R_1 \sqrt{\frac{6 m \rho S L^2 \omega_a^2 \alpha_a^2 \alpha_0^2}{R_1 \left(3 - \frac{3}{2} \alpha_0^2 - \frac{3}{8} \alpha_a^2\right)}} \quad (56)$$

Table 2 Model validation results: relative model prediction error for turning radius and turning period, when the tail beats at 1.5 Hz

(α_a, α_0)	Turning radius error (%)	Turning period error (%)
(15 deg, 45 deg)	11.23	8.80
(15 deg, 50 deg)	7.57	11.65
(20 deg, 45 deg)	8.46	3.82
(20 deg, 50 deg)	8.74	12.55
(25 deg, 45 deg)	2.90	0.97
(25 deg, 50 deg)	11.07	15.88

$$d_2 = \frac{2m_1}{\rho S} \sqrt{\frac{(\theta_1 + \theta_2 \alpha_0) m L^2 c \omega_a^2 \alpha_a^2 \alpha_0}{4}} \quad (57)$$

Using Eq. (55) and letting $c_1 = (1/c_0)$, we can obtain the following:

$$y = c_1^2 d_2^2 + d_1^2 C_D^2 c_1^2 + 2c_1 d_2 d_1 C_D \quad (58)$$

where $y = R^2 |\bar{x}_3|^2 - \bar{x}_1^2$. Letting $\phi_1 = c_1^2$, $\phi_2 = C_D^2 c_1^2$, and $\phi_3 = c_1^2 C_D$, one can rewrite the previous expression as

$$y = \phi_1 d_2^2 + d_1^2 \phi_2 + 2d_2 d_1 \phi_3 \quad (59)$$

With a set of collected data, the parameters ϕ_1 through ϕ_3 can be estimated readily using techniques, such as the constrained linear-least square method. We can then solve for C_D , C_L , and K_f using the definitions established previously. Since the proposed estimation method only provides the ratio (K_m/K_D), to obtain the values for K_m and K_D , we run simulations with the original dynamical model and choose K_D such that the angular velocity of the dynamic model matches that of the averaged model.

For the implementation of the earlier parameter estimation scheme, we first ran experiments to obtain the steady-state turning radii and periods for different tail biases (0 deg, 25 deg, 40 deg), and amplitudes (15 deg, 20 deg, 25 deg) while holding the frequency at 1 Hz. The values obtained for the parameters K_f , K_m , C_D , C_L , and K_D are listed in Table 1. Furthermore, to validate the models we ran experiments with the same set of biases and amplitudes as previously stated while holding the frequency at 1.5 Hz. Table 2 lists the errors in turning radius and period between those obtained from experiments and those obtained from simulation using the parameters estimated earlier. The comparison indicates that the estimated model has acceptable accuracy.

5.2 Experimental Results on Path-Following. The parameters used to solve the optimization problem and implement the NMPC were as follows:

- Length of optimization horizon : $T_c = T_p = 7s$
- Sampling interval : $t_s = 1s$
- Weighting matrix : $Q = 0.9I_5$
- Control weighting matrix : $R = 0.001I_3$
- $V_{c \max} = 0.04$ m/s
- $\dot{s}_{\max} = 0.04$ m/s
- $\alpha_{0 \min} = -40$ deg
- $\alpha_{0 \max} = 40$ deg
- $\alpha_{a \max} = 30$ deg
- $\alpha_{a \min} = 0$ deg

The following were the inputs constraints used for the case implementing projection:

$$\begin{bmatrix} -1.810 & \leq u_{e1} \leq 0.190 \\ 0 & \leq u_{f1} \leq 0.794 \\ -0.191 & \leq u_{f2} \leq 0.191 \end{bmatrix} \quad (60)$$

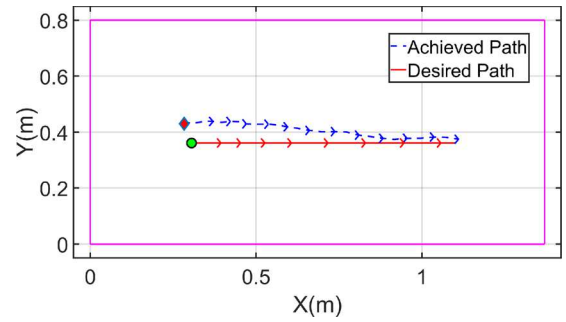


Fig. 15 Experiments: line-tracking results for NMPC with the proposed control projection method

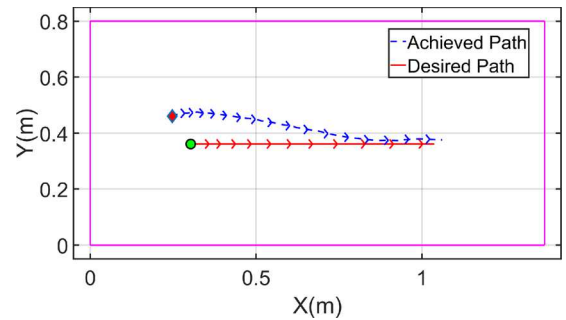


Fig. 16 Experiments: line-tracking results for NMPC with boxed constraints inside the admissible sector-shaped control region

For the case using boxed constraints (without projection), we considered the following input constraints:

$$\begin{bmatrix} -1.81 \leq u_{e1} \leq 0.1900 \\ 0.366 \leq u_{f1} \leq 0.794 \\ -0.115 \leq u_{f2} \leq 0.115 \end{bmatrix} \quad (61)$$

We first considered the following path:

$$\begin{aligned} x_p &= s \\ y_p &= 0 \end{aligned} \quad (62)$$

where x_p and y_p represent the position of the point P in the {I} frame. The desired velocity for the robotic fish was set to be 0.03 m/s.

In Figs. 15 and 16 we compare the desired path and the closed-loop robotic fish trajectory, obtained by using the NMPC with the proposed control projection scheme, and with a boxed constraint inside the nonlinear constraint set U , respectively. We do not report the case of NMPC with nonlinear constraints U directly, because it could not be implemented in real-time due to its long computation time.

Similarly, we considered the following circular path:

$$\begin{aligned} x_p &= 0.3 \sin(s) \\ y_p &= 0.3 \cos(s) \end{aligned} \quad (63)$$

and Figs. 17 and 18 show the path-following results for NMPC with the proposed projection and for NMPC with boxed constraints inside U , respectively.

Overall, the tracking results shown in Figs. 15–18, one can see that, consistent with the simulation results, the proposed NMPC scheme with projection resulted in faster convergence to the

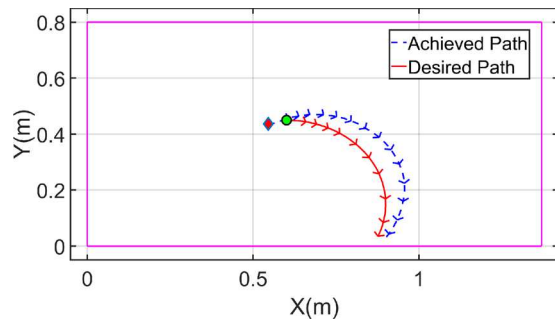


Fig. 17 Experiments: arc-tracking results for NMPC with the proposed control projection method

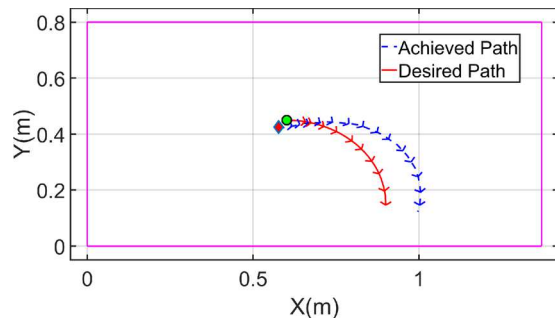


Fig. 18 Experiments: arc-tracking results for NMPC with boxed constraints inside the admissible sector-shaped control region

desired path and smaller path error, due to the availability of larger control authority.

6 Conclusion

In this paper, we proposed and implemented in real-time a path-following NMPC scheme for a tail-actuated robotic fish. A high-fidelity averaged nonlinear dynamic model was used for controller design. A parameter estimation scheme was employed to empirically identify the hydrodynamic parameters and scaling coefficients of the model. Furthermore, given that the control inputs were functions of two of the tail-beat parameters, specifically the tail bias and tail amplitude, a control projection strategy was implemented to handle these nonlinear input constraints and maximize the use of the admissible control region in a computationally efficient manner. Finally, simulation and experimental results demonstrated the effectiveness of the proposed scheme.

For future work, the proposed NMPC algorithm will be evaluated in an environmental sensing application, where there will be an upper-level path planning scheme integrated with the NMPC-based path-tracking scheme. Furthermore, in another direction, we plan to extend this work to robotic fish with more sophisticated dynamics, such as robotic fish actuated by both pectoral fins and caudal fin [18], and underwater robots like the gliding robotic fish [55].

Funding Data

- National Science Foundation (Grant Nos. DGE1424871, ECCS 1446793, and IIS 1715714; Funder ID: 10.13039/1000000001).

References

- [1] Tan, X., 2011, "Autonomous Robotic Fish as Mobile Sensor Platforms: Challenges and Potential Solutions," *Mar. Technol. Soc. J.*, **45**(4), pp. 31–40.

- [2] Triantafyllou, M., and Triantafyllou, G. S., 1995, "An Efficient Swimming Machine," *Sci. Am.*, **272**(3), pp. 64–70.
- [3] Guo, S., Fukuda, T., and Asaka, K., 2003, "A New Type of Fish-Like Underwater Microbot," *IEEE/ASME Trans. Mechatronics*, **8**(1), pp. 136–141.
- [4] Hu, H., Liu, J., Dukes, I., and Francis, G., 2006, "Design of 3D Swim Patterns for Autonomous Robotic Fish," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Beijing, China, Oct. 9–15, pp. 2406–2411.
- [5] Costa, D., Franciolini, M., Palmieri, G., Crivellini, A., and Scaradozzi, D., 2017, "Computational Fluid Dynamics Analysis and Design of an Ostraciiform Swimming Robot," *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Macau, China, Dec. 5–8, pp. 135–140.
- [6] Chen, X., Yu, J., Wu, Z., Meng, Y., and Kong, S., 2018, "Toward a Maneuverable Miniature Robotic Fish Equipped With a Novel Magnetic Actuator System," *IEEE Trans. Syst. Man Cybern. Syst.*, **13**(3), pp. 1–11.
- [7] Liao, P., Zhang, S., and Sun, D., 2018, "A Dual Caudal-Fin Miniature Robotic Fish With an Integrated Oscillation and Jet Propulsive Mechanism," *Bioinspiration Biomimetics*, **13**(3), p. 036007.
- [8] Alessi, A., Sudano, A., Accoto, D., and Guglielmelli, E., 2012, "Development of an Autonomous Robotic Fish," *Fourth IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechanics (BioRob)*, Rome, Italy, June 24–27, pp. 1032–1037.
- [9] Ichikizaki, T., and Yamamoto, I., 2007, "Development of Robotic Fish With Various Swimming Functions," *Symposium on Underwater Technology and Workshop on Scientific Use of Submarine Cables and Related Technologies*, Tokyo, Japan, Apr. 17–20, pp. 378–383.
- [10] Low, K., and Willy, A., 2005, "Development and Initial Investigation of NTU Robotic Fish With Modular Flexible Fins," *IEEE International Conference Mechatronics and Automation (ICMA)*, Niagara Falls, ON, Canada, July 29–Aug. 1, pp. 958–963.
- [11] Tan, X., Kim, D., Usher, N., Laboy, D., Jackson, J., Kapetanovic, A., Rapai, J., Sabadus, B., and Zhou, X., 2006, "An Autonomous Robotic Fish for Mobile Sensing," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Beijing, China, Oct. 9–15, pp. 5424–5429.
- [12] Morgansen, K. A., Triplett, B. I., and Klein, D. J., 2007, "Geometric Methods for Modeling and Control of Free-Swimming Fin-Actuated Underwater Vehicles," *IEEE Trans. Rob.*, **23**(6), pp. 1184–1199.
- [13] Chen, X., Shatara, S., and Tan, X., 2010, "Modeling of Biomimetic Robotic Fish Propelled by an Ionic Polymer-Metal Composite Caudal Fin," *IEEE/ASME Trans. Mechatronics*, **15**(3), pp. 448–459.
- [14] Kelly, S., and Murray, R., 2000, "Modeling Efficient Pisciform Swimming for Control," *Int. J. Robust Nonlinear Control*, **10**(4), pp. 217–241.
- [15] Wang, J., Alequin-Ramos, F., and Tan, X., 2011, "Dynamic Modeling of Robotic Fish and Its Experimental Validation," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, San Francisco, CA, Sept. 25–30, pp. 588–594.
- [16] Wang, J., McKinley, P., and Tan, X., 2012, "Dynamic Modeling of Robotic Fish With a Flexible Caudal Fin," *ASME Paper No. DSCC2012-MOVIC2012-8695*.
- [17] Behbahani, S. B., and Tan, X., 2016, "Bio-Inspired Flexible Joints With Passive Feathering for Robotic Fish Pectoral Fins," *Bioinspiration Biomimetics*, **11**(3), p. 036009.
- [18] Kopman, V., Laut, J., Acquaviva, F., Rizzo, A., and Porfiri, M., 2015, "Dynamic Modeling of a Robotic Fish Propelled by a Compliant Tail," *IEEE J. Oceanic Eng.*, **40**(1), pp. 209–221.
- [19] Barrett, D., Grosenbaugh, M., and Triantafyllou, M., 1996, "The Optimal Control of a Flexible Hull Robotic Undersea Vehicle Propelled by an Oscillating Foil," *Autonomous Underwater Vehicle Technology (AUV)*, Monterey, CA, June 2–6, pp. 1–9.
- [20] Anderson, J. M., and Chhabra, N. K., 2002, "Maneuvering and Stability Performance of a Robotic Tuna," *Integr. Comp. Biol.*, **42**(1), pp. 118–126.
- [21] Wang, S., Zhang, Z., and Sang, H., 2004, "Analysis of Velocity Control Algorithms for Biomimetic Robot Fish," *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Shenyang, China, Aug. 22–26, pp. 972–976.
- [22] Zhou, C., Cao, Z., Wang, S., and Tan, M., 2006, "The Posture Control and 3-D Locomotion Implementation of Biomimetic Robot Fish," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Beijing, China, Oct. 9–15, pp. 5406–5411.
- [23] Hirata, K., Takimoto, T., and Tamura, K., 2000, "Study on Turning Performance of a Fish Robot," *First International Symposium on Aqua Bio-Mechanisms*, Honolulu, HI, pp. 287–292.
- [24] Morgansen, K. A., Duidam, V., Mason, R. J., Burdick, J. W., and Murray, R. M., 2001, "Nonlinear Control Methods for Planar Carangiform Robot Fish Locomotion," *International Conference on Robotics and Automation (ICRA)*, Seoul, South Korea, May 21–26, pp. 427–434.
- [25] Yu, J., Wang, L., and Tan, M., 2005, "A Framework for Biomimetic Robot Fish's Design and Its Realization," *American Control Conference (ACC)*, Portland, OR, June 8–10, pp. 1593–1598.
- [26] Zhang, F., Zhang, F., and Tan, X., 2014, "Tail-Enabled Spiraling Maneuver for Gliding Robotic Fish," *ASME J. Dyn. Syst., Meas., Control*, **136**(4), p. 041028.
- [27] Makrodimitis, M., Nanos, K., and Papadopoulos, E., 2017, "A Novel Trajectory Planning Method for a Robotic Fish," *25th Mediterranean Conference on Control and Automation (MED)*, Valletta, Malta, July 3–6, pp. 1119–1124.
- [28] Wang, L., Wang, S., Cao, Z., Tan, M., Zhou, C., Sang, H., and Shen, Z., 2005, "Motion Control of a Robot Fish Based on CPG," *IEEE International Conference on Industrial Technology (ICIT)*, Hong Kong, China, Dec. 14–17, pp. 1263–1268.
- [29] Zhao, W., Hu, Y., Zhang, L., and Wang, L., 2009, "Design and CPG-Based Control of Biomimetic Robotic Fish," *IET Control Theory Appl.*, **3**(3), pp. 281–293.

- [30] Wang, W., Guo, J., Wang, Z., and Xie, G., 2013, "Neural Controller for Swimming Modes and Gait Transition on an Ostraciiform Fish Robot," *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)* Wollongong, Australia, July 9–12, pp. 1564–1569.
- [31] Hu, Y., Tian, W., Liang, J., and Wang, T., 2011, "Learning Fish-Like Swimming With a CPG-Based Locomotion Controller," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, San Francisco, CA, Sept. 25–30, pp. 1863–1868.
- [32] Ren, Q., Xu, J., and Li, X., 2015, "A Data-Driven Motion Control Approach for a Robotic Fish," *J. Bionic Eng.*, **12**(3), pp. 382–394.
- [33] Wang, W., Gu, D., and Xie, G., 2017, "Autonomous Optimization of Swimming Gait in a Fish Robot With Multiple Onboard Sensors," *IEEE Trans. Syst. Man, Cybern. Syst.*, (epub).
- [34] Morgansen, K. A., Vela, P. A., and Burdick, J. W., 2002, "Trajectory Stabilization for a Planar Carangiform Robot Fish," *IEEE International Conference on Robotics and Automation (ROBOT)*, Washington, DC, May 11–15, pp. 756–762.
- [35] Yu, J., Tan, M., Wang, S., and Chen, E., 2004, "Development of a Biomimetic Robotic Fish and Its Control Algorithm," *IEEE Trans. Syst. Man, Cybern. Part B (Cybern.)*, **34**(4), pp. 1798–1810.
- [36] Saimek, S., and Li, P. Y., 2004, "Motion Planning and Control of a Swimming Machine," *Int. J. Rob. Res.*, **23**(1), pp. 27–53.
- [37] Hu, Y., Zhao, W., and Wang, L., 2009, "Vision-Based Target Tracking and Collision Avoidance for Two Autonomous Robotic Fish," *IEEE Trans. Ind. Electron.*, **56**(5), pp. 1401–1410.
- [38] Kato, N., 2000, "Control Performance in the Horizontal Plane of a Fish Robot With Mechanical Pectoral Fins," *IEEE J. Oceanic Eng.*, **25**(1), pp. 121–129.
- [39] Zou, K., Wang, C., Xie, G., Chu, T., Wang, L., and Jia, Y., 2009, "Cooperative Control for Trajectory Tracking of Robotic Fish," *American Control Conference (ACC)*, St. Louis, MO, June 10–12, pp. 5504–5509.
- [40] Suebsaiprom, P., and Lin, C.-L., 2015, "Maneuverability Modeling and Trajectory Tracking for Fish Robot," *Control Eng. Pract.*, **45**, pp. 22–36.
- [41] Suebsaiprom, P., and Lin, C.-L., 2016, "Sliding Mode Path Tracking Control for Fish-Robot Under Ocean Current Perturbation," *12th IEEE International Conference on Control and Automation (ICCA)*, Kathmandu, Nepal, June 1–3, pp. 972–977.
- [42] Gao, F. D., Pan, C., Han, Y., and Zhang, X., 2012, "Nonlinear Trajectory Tracking Control of a New Autonomous Underwater Vehicle in Complex Sea Conditions," *J. Central South Univ.*, **19**(7), pp. 1859–1868.
- [43] Xiang, X., Lapiere, L., Liu, C., and Jouvencel, B., 2011, "Path Tracking: Combined Path Following and Trajectory Tracking for Autonomous Underwater Vehicles," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, San Francisco, CA, Sept. 25–30, pp. 3558–3563.
- [44] Dongbing, G., and Huosheng, H., 2005, "A Stabilizing Receding Horizon Regulator for Nonholonomic Mobile Robots," *IEEE Trans. Rob.*, **21**(5), pp. 1022–1028.
- [45] Kanjanawanishkul, K., and Zell, A., 2009, "Path Following for an Omnidirectional Mobile Robot Based on Model Predictive Control," *IEEE International Conference on Robotics and Automation (ROBOT)*, Kobe, Japan, May 12–17, pp. 3341–3346.
- [46] Li, X., Kanjanawanishkul, K., and Zell, A., 2008, "Nonlinear Model Predictive Control of an Omnidirectional Mobile Robot," *Tenth International Conference on Intelligent Autonomous Systems (IAS 2008)*, Vigo, Spain, June 4–7, pp. 92–99.
- [47] Kanjanawanishkul, K., 2015, "MPC-Based Path Following Control of an Omnidirectional Mobile Robot With Consideration of Robot Constraints," *Adv. Electr. Electron. Eng.*, **13**(1), pp. 54–63.
- [48] Wang, J., and Tan, X., 2015, "Averaging Tail-Actuated Robotic Fish Dynamics Through Force and Moment Scaling," *IEEE Trans. Rob.*, **31**(4), pp. 906–917.
- [49] Ariens, D., Houska, B., Ferreau, H., and Acado, L. F., "Toolkit for Automatic Control and Dynamic Optimization," Leuven, Belgium, accessed Mar. 22, 2019, <http://acado.github.io>
- [50] Castano, M. L., and Tan, X., 2016, "Model Predictive Control of a Tail-Actuated Robotic Fish," *ASME Paper No. DSCC 2016-9918*.
- [51] Garcia, C. E., Prett, D. M., and Morari, M., 1989, "Model Predictive Control: Theory and Practice—A Survey," *Automatica*, **25**(3), pp. 335–348.
- [52] Allgöwer, F., Findeisen, R., and Nagy, Z. K., 2004, "Nonlinear Model Predictive Control: From Theory to Application," *J. Chin. Inst. Chem. Eng.*, **35**(3), pp. 299–315.
- [53] Yu, S., Li, X., Chen, H., and Allgower, F., 2012, "Nonlinear Model Predictive Control for Path Following Problems," *IFAC Proc. Vol.*, **45**(17), pp. 145–150.
- [54] Aureli, M., Kopman, V., and Porfiri, M., 2010, "Free-Locomotion of Underwater Vehicles Actuated by Ionic Polymer Metal Composites," *IEEE/ASME Trans. Mechatronics*, **15**(4), pp. 603–614.
- [55] Zhang, F., Ennasr, O., Litchman, E., and Tan, X., 2016, "Autonomous Sampling of Water Columns Using Gliding Robotic Fish: Algorithms and Harmful-Algae-Sampling Experiments," *IEEE Syst. J.*, **10**(3), pp. 1271–1281.