Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnline CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 58-20 (2024) 253–258

Nonlinear model predictive control for sinusoidal gait tracking for an underwater snake robot

Amer Orucevic†, Eirik Lothe Foseid†, Mads Erlend Bøe Lysø, Kristin Ytterstad Pettersen, Jan Tommy Gravdahl.

Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), Trondheim, Norway (e-mail: {amer.orucevic, eirik.l.foseid, mads.e.b.lyso, kristin.y.pettersen, jan.tommy.gravdahl}@ntnu.no)

Abstract:
Energy efficiency is crucial for the operational time and reach of autonomous underwater vehicles (AUVs). A new class of AUVs, underwater snake robots (USRs), has an articulated body that may be utilized to enhance propulsion efficiency and achieve energy autonomy. This paper applies nonlinear model predictive control (NMPC) to achieve sinusoidal gait tracking for underwater snake robots (USRs). We present a comprehensive simulation study that incorporates high-fidelity modeling of fluid-structure interaction, which validates the control design. Additionally, we showcase the potential for significant energy savings by fine-tuning the cost function, leading to reduced actuator power consumption.

Keywords: Nonlinear and optimal marine system control, Energy Autonomy, Robotics technology, Autonomous robotic systems, Autonomous underwater vehicles.

1. INTRODUCTION

Exploring the depths of the oceans is essential for addressing various challenges, including ensuring an adequate food supply, preserving biodiversity, harnessing renewable energy, enabling efficient transport, and accessing valuable minerals and resources. To fully explore the vast expanses of the ocean, we require advanced marine robots that are both efficient and autonomous.

One promising solution is the utilization of underwater snake robots (USRs), which are autonomous underwater vehicles (AUVs) composed of interconnected slender segments with revolute joints. This unique design enables the USRs to move by undulating, similar to the motion of an eel (Kelasidi et al., 2016). Their slender cross-section also allows them to navigate through narrow spaces while their articulated nature enables them to interact with the surrounding environment, effectively functioning as robotic manipulators.

However, one major challenge for AUVs is the power supply. The limited capacity of batteries restricts their operational time. The use of tethers, as for remotely operated vehicles (ROVs), would, however, impose limitations on their range and autonomy. Consequently, there is a need to explore alternative methods to enhance propulsion efficiency and achieve energy autonomy for AUVs. One intriguing approach that the articulated body of USRs facilitates, involves harnessing the motion generated in the joints when the USR operates in the wake of a bluff body (Bernier et al. (2019, 2018)). In this paper we investigate this concept further to determine how nonlinear model predictive control (NMPC) can be applied to accurately track a sinusoidal swimming pattern while reducing the energy consumption of the actuators. Furthermore, we will compare the performance and energy consumption of the USR under different tuning configurations.

Model predictive control (MPC) is an optimization-based control method that relies on mathematical models of a system to make predictions and optimize control actions over a finite time horizon. The control input sequence that minimizes a specified cost function is determined by repeatedly solving a constrained optimization problem. The first control action in the optimal sequence is applied to the system, and the process is then repeated for each time step. The research on MPC is extensive; see, e.g., Rawlings et al. (2017) for more information. The NMPC extends the concept of MPC to nonlinear systems by handling nonlinear system dynamics and system constraints, cf. Findeisen and Allgöwer (2002).

The application of MPC to land-based snake robots has been studied in several papers. A simplified land-based snake robot model was presented in Fukushima et al. (2021) using the averaging method. An MPC was then used with the simplified model for path-following. Additionally, soft constraints were introduced to reduce the effects of approximation errors. However, this approach

⋆ This result is part of a project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme, through the ERC Advanced Grant 101017697-CRÈME. The work is also supported by the Research Council of Norway through the Centres of Excellence funding scheme, project No. 223254 – NTNU AMOS.

† These authors contributed equally to this work.

10.1016/j.ifacol.2024.10.063

considers a linearized model of the snake robot, which may result in significant differences between the model and the actual system with a corresponding degradation in the controller performance.

More complex nonlinear models of land-based snake robots are considered in Marafioti et al. (2014), where an NMPC is applied for straight-line path following. The controller generates a turning offset to the undulating gait pattern, which allows the snake robot to track a path. Nonlinear models of land-based snake robots are also considered in Nonhoff et al. (2019), where an economic MPC is designed to integrate gait generation while maximizing the forward velocity. The resulting controller is shown to outperform a standard lateral undulation controller, simultaneously achieving higher forward velocity and lower energy use. Furthermore, both planar motion and obstacle-aided locomotion are assessed. However, the approach considers a simplified control oriented model of the USR, which may impact controller performance adversely. Moreover, the robustness of the MPC method to model uncertainties remains to be studied.

In this paper, we investigate how an NMPC approach can be applied to snake robots that operate underwater for more efficient movement. To this end, we implement an NMPC scheme to track a sinusoidal gait with the goal of achieving forward propulsion. Moreover, we investigate how tuning the cost function affects the energy usage. A simulation study using a high-fidelity simulator that captures fluid-structure interaction is presented, and the results show that the USR converges to the desired sinusoidal gait.

The paper is organized as follows: In Section 2, the methods used are presented. First, an overview of the simulation method and models we use are presented. Then, the application of an NMPC method to a USR and the energy used by the actuators are discussed. In Section 3, the simulation setup and results are presented and discussed. Finally, in Section 4, thoughts about future work and a conclusion are given.

2. METHODS

2.1 Coupled Solver

This section briefly summarizes the algorithm simulating a planar articulated swimmer in a complex fluid environment with fluid-structure interaction. The method is presented in Bernier et al. (2019) and relies on vortex particle-mesh (VPM) techniques coupled with a multi-body system solver (MBS). The algorithm consists of a computational fluid dynamics (CFD) component that handles the fluid dynamics, fluid-structure interaction, and placement of the swimmer in the fluid and a MBS that handles the control and dynamics of the swimmer, as illustrated in Fig. 1.

The VPM method solves the incompressible flow past deforming objects by using the velocity-vorticity formulation of the Navier-Stokes equations,

$$\frac{D\omega_f}{Dt} = (\omega_f \cdot \nabla)\boldsymbol{u}_f + \nu\nabla^2\omega_f, \tag{1a}$$

where D/Dt denotes the Lagrangian derivative, $\boldsymbol{u}_f$ is the velocity field, ν is the kinematic viscosity and ω_f is the vorticity field. The forces and moments affecting the swimmer, given by $\boldsymbol{F}_{\text{hyd}} = [\boldsymbol{F}_x, \boldsymbol{F}_y]^\top$ and $\boldsymbol{\tau}_{\text{hyd}}$, are calculated through the CFD, and are then sent to the MBS. For more detail on the CFD, see Bernier et al. (2019). The control input for the swimmer is then calculated in the MBS and applied for the current time step. The dynamics of the swimmer are then calculated and integrated to find the global position $\boldsymbol{p}_{\text{cm}} = [p_{\text{cm},x}, p_{\text{cm},y}]$ of the center of mass of the swimmer, and the global orientation of each link $\boldsymbol{\theta} = [\theta_1, \theta_2, \ldots, \theta_{N_l}]^\top$, where N_l is the number of links.

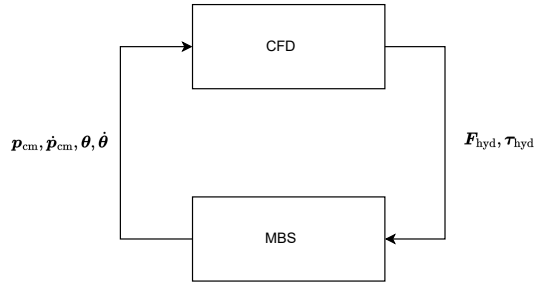


Fig. 1. Coupled solver

2.2 Multi-body System Solver

The MBS is presented in this section, with the equations of motion for a USR. A detailed model of a USR, where the full kinematics and dynamics of a planar snake robot with revolute joints are considered, is presented in Kelasidi et al. (2014).

We denote the orientation and position along the x- and y-axis of link i as θ_i, $p_{x,i}$ and $p_{y,i}$, respectively. The forces along the x- and y-axis, and moments acting on link i, generated by the CFD, are denoted $f_{hx,i}$, $f_{hy,i}$ and $\tau_{h,i}$, respectively. Furthermore, we define $\boldsymbol{F}_x = [f_{hx,1}, f_{hx,2}, ..., f_{hx,N_l}]$ and $\boldsymbol{F}_y = [f_{hy,1}, f_{hy,2}, ..., f_{hy,N_l}]$.

The equations of motion are given by

$$\boldsymbol{M}_\theta\ddot{\boldsymbol{\theta}} + \boldsymbol{W}_\theta\dot{\boldsymbol{\theta}}^2 + \boldsymbol{Q}_\theta(\boldsymbol{F}) + \boldsymbol{\tau}_{\text{hyd}} = -\boldsymbol{D}^T\boldsymbol{u}, \tag{2a}$$

$$\ddot{\boldsymbol{p}}_{\text{cm}} = \frac{1}{N_l m}\begin{bmatrix}\boldsymbol{e}^T\boldsymbol{F}_x \\ \boldsymbol{e}^T\boldsymbol{F}_y\end{bmatrix}, \tag{2b}$$

where

$$\boldsymbol{M}_\theta = J\boldsymbol{I}_{Nl} + ml^2\boldsymbol{S}_\theta\boldsymbol{V}\boldsymbol{S}_\theta + ml^2\boldsymbol{C}_\theta\boldsymbol{V}\boldsymbol{C}_\theta, \tag{3a}$$

$$\boldsymbol{W}_\theta = ml^2\boldsymbol{S}_\theta\boldsymbol{V}\boldsymbol{C}_\theta - ml^2\boldsymbol{C}_\theta\boldsymbol{V}\boldsymbol{S}_\theta, \tag{3b}$$

$$\boldsymbol{Q}_\theta(\boldsymbol{F}) = \begin{bmatrix}-l\boldsymbol{S}_\theta\boldsymbol{K}\boldsymbol{F}_x \\ l\boldsymbol{C}_\theta\boldsymbol{K}\boldsymbol{F}_y\end{bmatrix}, \tag{3c}$$

$$\boldsymbol{V} = \boldsymbol{A}^T(\boldsymbol{D}\boldsymbol{D}^T)^{-1}\boldsymbol{A}, \tag{3d}$$

$$\boldsymbol{K} = \boldsymbol{A}^T(\boldsymbol{D}\boldsymbol{D}^T)^{-1}\boldsymbol{D}. \tag{3e}$$

The matrices with the subscript θ depend on $\boldsymbol{\theta}$; the functional argument is omitted to save space. The equations are in the same form as presented in Kelasidi et al. (2016) for USRs. However, terms representing hydrodynamical effects in the model found in Kelasidi et al. (2016) are here replaced with forces and moments from the CFD simulation, in order to capture the interaction between the fluid and swimmer. The mass, moment of inertia, and half the length of the links are given by m, J, and l, respectively. The relative angles between the links are defined as

$$\phi_i = \theta_i - \theta_{i+1}, \tag{4}$$

we also define the vector $\boldsymbol{\phi} = [\phi_1, \phi_2, \ldots, \phi_{N_l-1}]^\top$. The transformation between the absolute orientation and relative angles is given by

$$\boldsymbol{\phi} = \boldsymbol{H}_\phi \boldsymbol{\theta}, \tag{5}$$

where

$$\boldsymbol{H}_\phi = \begin{bmatrix} 1 & -1 & 0 & \ldots & 0 & 0 \\ 0 & 1 & -1 & \ldots & 0 & 0 \\ \vdots & & \ddots & & \vdots & \\ 0 & 0 & 0 & \ldots & 1 & -1 \end{bmatrix} \in \mathbb{R}^{(N_l-1)\times N_l}. \tag{6}$$

The position of the center of mass of each link can be found through the transformation

$$\boldsymbol{p} = \begin{bmatrix} \boldsymbol{p}_x \\ \boldsymbol{p}_y \end{bmatrix} = \begin{bmatrix} -l\boldsymbol{K}^T \cos\boldsymbol{\theta} + \boldsymbol{e}p_{\mathrm{cm},x} \\ -l\boldsymbol{K}^T \sin\boldsymbol{\theta} + \boldsymbol{e}p_{\mathrm{cm},y} \end{bmatrix}, \tag{7}$$

where $\boldsymbol{p}_x = [p_{x,1}, p_{x,2}, \ldots, p_{x,N_l}]^T$ and $\boldsymbol{p}_y = [p_{y,1}, p_{y,2}, \ldots, p_{y,N_l}]$.

2.3 Nonlinear Model Predictive Control for USRs

This section discusses how the NMPC was implemented to achieve the USR tracking a sinusoidal gait. The equations of motion of the USR can be written as

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)), \tag{8}$$

where $\boldsymbol{x} = [\boldsymbol{p}_{\mathrm{cm}}, \dot{\boldsymbol{p}}_{\mathrm{cm}}, \boldsymbol{\theta}, \dot{\boldsymbol{\theta}}]^\top$ and $\boldsymbol{f}$ is given by (2). To achieve a computationally tractable prediction model, we assume the USR will operate in calm water with no current and moving slowly, so the drag forces will be small. Therefore, we set $\boldsymbol{F}_{\mathrm{hyd}} = \boldsymbol{\tau}_{\mathrm{hyd}} = 0$, and assume no knowledge of the hydrodynamical forces in the prediction model. To represent the NMPC design as an optimization problem, the equations of motion (8) are discretized. We define

$$\boldsymbol{x}_{k+1} = \boldsymbol{F}(\boldsymbol{x}_k, \boldsymbol{u}_k), \tag{9}$$

where $\boldsymbol{F}(\cdot)$ integrates the system dynamics over the given time step. We use a 4th order Runge-Kutta method given by

$$\boldsymbol{k}_1 = f(\boldsymbol{x}_k, \boldsymbol{u}_k), \tag{10a}$$

$$\boldsymbol{k}_2 = f(\boldsymbol{x}_k + \frac{dt}{2}\boldsymbol{k}_1, \boldsymbol{u}_k), \tag{10b}$$

$$\boldsymbol{k}_3 = f(\boldsymbol{x}_k + \frac{dt}{2}\boldsymbol{k}_2, \boldsymbol{u}_k), \tag{10c}$$

$$\boldsymbol{k}_4 = f(\boldsymbol{x}_k + dt\boldsymbol{k}_3, \boldsymbol{u}_k), \tag{10d}$$

$$\boldsymbol{x}_{k+1} = \frac{dt}{6}(\boldsymbol{k}_1 + 2\boldsymbol{k}_2 + 2\boldsymbol{k}_3 + \boldsymbol{k}_4), \tag{10e}$$

where dt is the time-step. We formulate the NMPC problem for the USR as

$$\min_{\bar{\boldsymbol{u}}_k} \sum_{j=0}^{N} \varphi(\boldsymbol{x}_{k+j}, \boldsymbol{\phi}^{\mathrm{ref}}_{k+j}, \boldsymbol{u}_{k+j}), \tag{11a}$$

Subject to:

$$\boldsymbol{x}_k = \boldsymbol{x}(k) \tag{11b}$$

$$\boldsymbol{x}_{k+j+1} = \boldsymbol{F}(\boldsymbol{x}_{k+j}, \boldsymbol{u}_{k+j}), \quad j = 0, \ldots, N-1 \tag{11c}$$

$$|\boldsymbol{u}_{k+j}| \leq u_{\max}\boldsymbol{1}_{(N_l-1)\times 1}, \quad j = 0, \ldots, N-1 \tag{11d}$$

$$|\boldsymbol{x}_{k+j}| \leq x_{\max}\boldsymbol{1}_{(2N_l+4)\times 1}, \quad j = 0, \ldots, N-1 \tag{11e}$$

where $\bar{\boldsymbol{u}}_k = [\boldsymbol{u}_k^\top, \ldots, \boldsymbol{u}_{k+N-1}^\top]^\top$ is the control sequence vector containing the control input vector for each time step, and N is the prediction horizon. The initial value $\boldsymbol{x}_k$ is set to the current state $\boldsymbol{x}(k)$ of the system, as seen in (11b). The prediction model is given by (11c). Furthermore, the input and state constraints are given by (11d) and (11e), respectively. The cost functions (11a) are given by

$$\begin{aligned} &\varphi(\boldsymbol{x}_k, \boldsymbol{\phi}^{\mathrm{ref}}_k, \boldsymbol{u}_k) = \\ &\quad (\boldsymbol{H}_x\boldsymbol{x}_k - \boldsymbol{\phi}^{\mathrm{ref}}_k)^\top \boldsymbol{Q}(\boldsymbol{H}_x\boldsymbol{x}_k - \boldsymbol{\phi}^{\mathrm{ref}}_k) + \boldsymbol{u}_k^\top \boldsymbol{R}\boldsymbol{u}_k, \end{aligned} \tag{12}$$

where $\boldsymbol{H}_x = [\boldsymbol{0}_{(N_l-1)\times 4}, \boldsymbol{H}_\phi, \boldsymbol{0}_{(N_l-1)\times N_l}]$. The references for the relative angles are given by $\boldsymbol{\phi}^{\mathrm{ref}}_k = [\phi^{\mathrm{ref}}_{k,1}, \ldots, \phi^{\mathrm{ref}}_{k,N_l-1}]^\top$, and

$$\phi^{\mathrm{ref}}_{k,i} = \alpha \sin(\omega(t + dt\cdot k) + \delta(i-1)) + \phi^{\mathrm{ref}}_{0,k}, \quad i = 1, \ldots, N_l - 1 \tag{13}$$

where α, ω, δ, and $\phi^{\mathrm{ref}}_{0,k}$ are the amplitude, frequency, offset and turning offset, which are design parameters for the swimming motion of the USR. The turning offset is used to control the turning rate of the USR Guo (2006); Kelasidi et al. (2014). The matrices $\boldsymbol{Q} \in \mathbb{R}^{(N_l-1)\times(N_l-1)}$ and $\boldsymbol{R} \in \mathbb{R}^{(N_l-1)\times(N_l-1)}$ are cost function weights. The overall goal of the controller is to track the given reference gait while reducing the amount of energy the actuators use.

2.4 Energy Consumption

The USR generates propulsion by moving its joints and interacting with the surrounding fluid, converting actuator torque into joint motion and fluid interaction. The energy spent for the propulsion of the robot is given by

$$E_s = \int_0^T \sum_{i=1}^{N_l-1} |u_i(t)\dot{\phi}_i(t)| dt, \tag{14}$$

where u_i and $\dot{\phi}_i$ are the input torque and relative angular velocity of link i, respectively, and the total swimming time is given by T. The average power consumption is given by

$$\bar{P}_{\mathrm{avg}} = \frac{1}{T} E_s. \tag{15}$$

Because the coupled solver operates in non-dimensional values, it is advantageous to consider a non-dimensionalized average power, which can be scaled to assess systems of different sizes. The non-dimensional average power is given by

$$\bar{P}^*_{\mathrm{avg}} = \frac{\bar{P}_{\mathrm{avg}}}{\rho_f L V^3_{Mt}}, \tag{16}$$

where ρ_f is the density of the fluid, L is the length of each link, and V_{Mt} is the maximal tangential velocity of the swimmer.

3. RESULTS

This section presents the results from a simulation study of the USR under the NMPC scheme presented in Section 2.3, using the coupled solver presented in Section 2.1.

3.1 Simulation Setup

The coupled solver and NMPC approach discussed in Section 2 are implemented in Python using the package CasADi by Andersson et al. (2018), which is an open-source tool for nonlinear optimization and algorithmic differentiation. The simulations are run for a USR with

$N_l = 5$. The Reynolds number is selected as $Re = 100$ for all simulations with a computational domain size given by $[0, 2.0] \times [0.0, 1.0]$ with a discretization grid resolution of $[512, 256]$. Furthermore, the kinematic viscosity is given by $\nu = \frac{V_{Mt}(2a)}{Re}$ and the maximal time step of the CFD simulation is defined as $\Delta t_{\max}$. No additional constraints were imposed on the swimmer states for the optimization problem, and the initial orientations of the links were set to 0. The swimmer dimensions are illustrated in Fig. 2, where the length and width from the center are given by a and b, respectively. The length from the tip of the ellipse to the revolute joint is given by c.

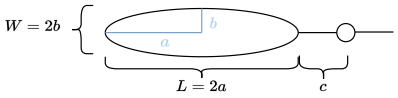


Fig. 2. Link and joint configuration.

The mass and moment of inertia of each link are calculated through

$$m = \rho_s \pi a b, \tag{17a}$$
$$J = (1/4)m(a^2 + b^2), \tag{17b}$$

where ρ_s is the mass density of the swimmer. The penalty matrix for the tracking error was set to $\boldsymbol{Q} = 5.0\boldsymbol{I}_{(N_l-1)}$, while the penalty matrix for large inputs was set to five different values, and the simulations were repeated to investigate how this affects the energy usage. The choices investigated were given by $\boldsymbol{R} = \gamma_R \boldsymbol{I}_{(N_l-1)}$, where the following values were used for γ_R in the simulations: $\gamma_R = [1.0, 2.0, 5.0, 10.0, 15.0]$. We also wish to investigate the effects of the turning offset in the reference model (13). Letting $\boldsymbol{\theta}_k$ be the predicted value of $\boldsymbol{\theta}$ at time k along the prediction horizon, we propose to use the turning offset presented by Liljebäck et al. (2013)

$$\phi_{0,k} = k_p \left(\bar{\boldsymbol{\theta}}_k - \bar{\theta}_k^{\text{ref}} \right) \tag{18}$$

where $\bar{\boldsymbol{\theta}}_k$ is the mean link orientation at time k. This term is used to steer the mean link orientation, also called the heading by Liljebäck et al. (2013), to a desired reference heading $\bar{\theta}_k^{\text{ref}}$. Additionally, the robustness of the method with respect to currents is investigated. To this end, a time-varying current given by

$$\boldsymbol{V}_c(t) = \begin{bmatrix} 0 \\ \frac{1}{2}\sin(0.1t) \end{bmatrix} \tag{19}$$

is introduced as an unmodeled disturbance in the simulation. The NMPC ran every $0.1s$, and the applied input was constant between each iteration. The remaining parameters used in the simulations are given in Table 1. See Fig. 3 for a snapshot of the simulation of the swimmer.

Fig. 3. Snapshot of the swimmer in the high-fidelity simulation, showing vorticity in a clockwise direction (red) and counter-clockwise direction (blue).

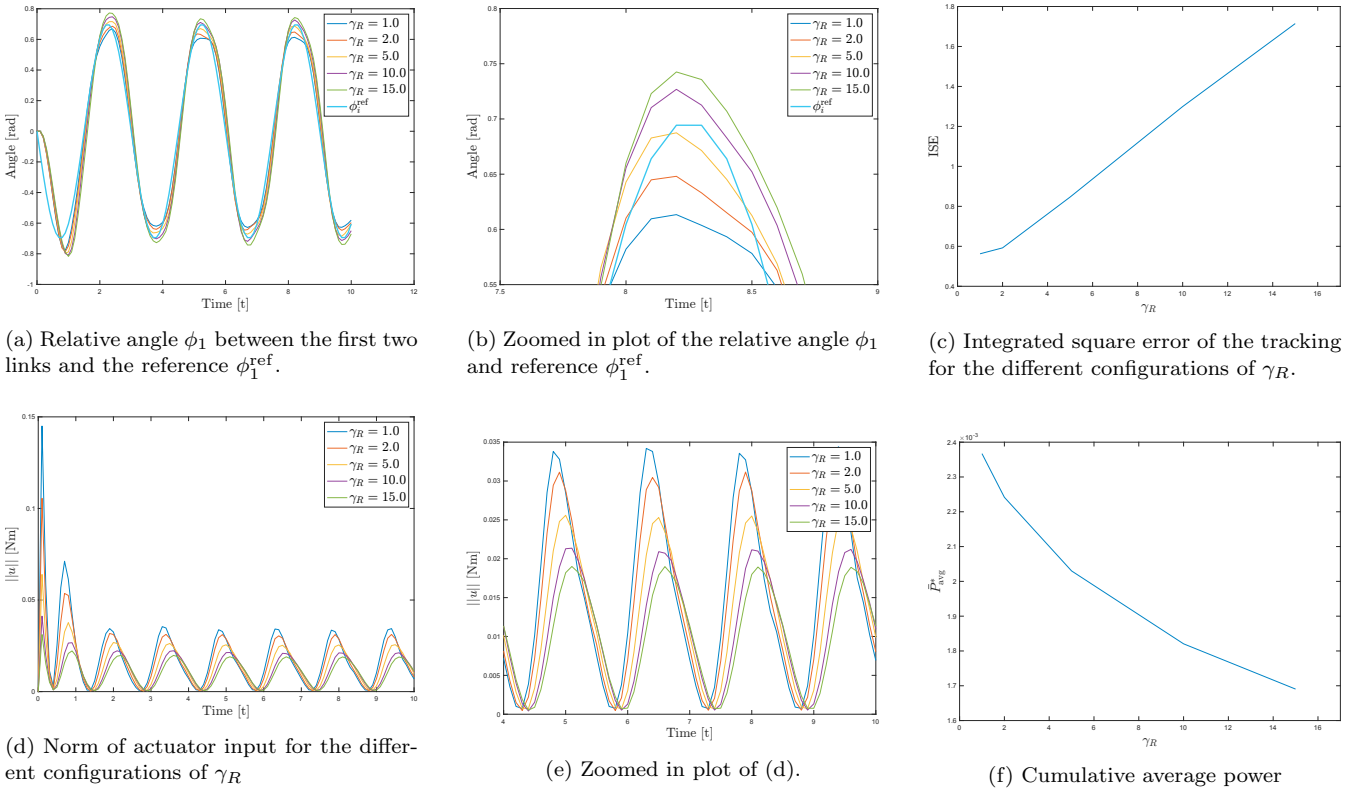
3.2 Performance

In this section, we present the results and discuss the performance of the NMPC approach. For the case without a turning offset, the tracking of the desired relative angle between the first two links is depicted in Fig. 4a for different configurations of γ_R. The USR successfully exhibits a swimming motion with the NMPC approach across all configurations. A zoomed-in plot in Fig. 4b reveals some variation in performance with increasing γ_R. Higher values tend to result in a more significant overshoot, while lower values undershoot the desired angle. However, these observations may be specific to this swimming pattern, and altering parameters like amplitude, frequency, or offset could yield different outcomes. Generally, lower γ_R values are expected to offer better tracking performance at the expense of higher control inputs. To assess performance more comprehensively, we calculate and plot the integral squared error (ISE) for each configuration in Fig. 4c. As γ_R increases, the error also increases. The norm of the actuator input, shown in Fig. 4d, exhibits spikes during the initial seconds but stabilizes in a periodic motion ranging between $0N$ and $0.35N$. A sharp decrease is observed with higher γ_R values. The average dimensionless power $\bar{P}^*_{\text{avg}}$ for each configuration, shown in Fig. 4, indicates that tuning the cost function results in a $\approx 30\%$ decrease in power usage. The findings highlight that there is a trade-off between tracking performance and energy consumption, as expected. Even with $\gamma_R = 15.0$, the USR achieves a swimming motion, although the prediction model used in the NMPC does not consider hydrodynamical forces and torques. Implementing more precise models of fluid-structure interaction could potentially enhance performance while optimizing energy usage for swimming motion.

For the case with a turning offset and time-varying currents, we start with an initial configuration of $\boldsymbol{\theta}_0 = [30^\circ\ 60^\circ\ 90^\circ\ 90^\circ\ 90^\circ]^T$, giving the initial heading angle $\overline{\boldsymbol{\theta}}_0 = 72^\circ$. We choose the reference heading angle $\overline{\theta}^{\text{ref}} = 0^\circ$. As can be seen in Fig. 6, the heading seems to converge to a neighborhood around the reference, where it oscillates for all values of γ_R. The amplitude of the oscillation increases as γ_R is increased. This is expected as large γ_R penalize the control input more leading to worse tracking performance, thus reducing the effect of the feedback term (18) driving the heading $\bar{\boldsymbol{\theta}}$ to the reference $\overline{\theta}^{\text{ref}}$. We also see from Fig. 5a

Table 1. Simulation parameters

V_{Mx}	l	a	b	c	ρ_f	ρ_s	dt	$\Delta t^{\max}$	$p_{x,0}$	$p_{y,0}$
0.24	$a+c$	0.03125	$0.2a$	0.01	997	997	0.1	0.001	1.0	0.5
T	$\boldsymbol{\theta}_0$	α	ω	δ	$u_{\max}$					
5.0	$\mathbf{0}_{(1\times N_l)}$	40°	120°	60°	0.5					



(a) Relative angle ϕ_1 between the first two links and the reference ϕ_1^{ref}.

(b) Zoomed in plot of the relative angle ϕ_1 and reference ϕ_1^{ref}.

(c) Integrated square error of the tracking for the different configurations of γ_R.

(d) Norm of actuator input for the different configurations of γ_R

(e) Zoomed in plot of (d).

(f) Cumulative average power

.

Fig. 4. Results from the high-fidelity simulation study without turning offset

and Fig. 5b that there is a large tracking error during the transient phase, which results from the change in initial configuration. This leads to a significantly higher control effort as seen in Fig. 5d, and resulting energy expenditure as seen in Fig. 5f.

The computational power required is a major drawback of the NMPC approach. In this study, the NMPC algorithm executed every $0.1s$, while the average time needed to compute the optimal solution per NMPC iteration was $0.45s$. However, it is expected that substantial improvements can be made with the use of more efficient code, and solvers, using frameworks such as Acados from Verschueren et al. (2021).

4. CONCLUSIONS AND FUTURE WORK

In this study, we employed the nonlinear model predictive control (NMPC) method to successfully generate a sinusoidal swimming pattern for underwater snake robot (USR), which are a new class of autonomous underwater vehicles (AUVs). We conducted high-fidelity simulations that accounted for fluid-structure interaction to evaluate its performance. Our findings demonstrate that despite not incorporating hydrodynamical forces and torques in the prediction model, the USR can achieve a sinusoidal swimming pattern in calm waters. Moreover, by fine-tuning the cost function to regulate the input, we significantly reduced the energy consumption required to achieve the desired swimming pattern.

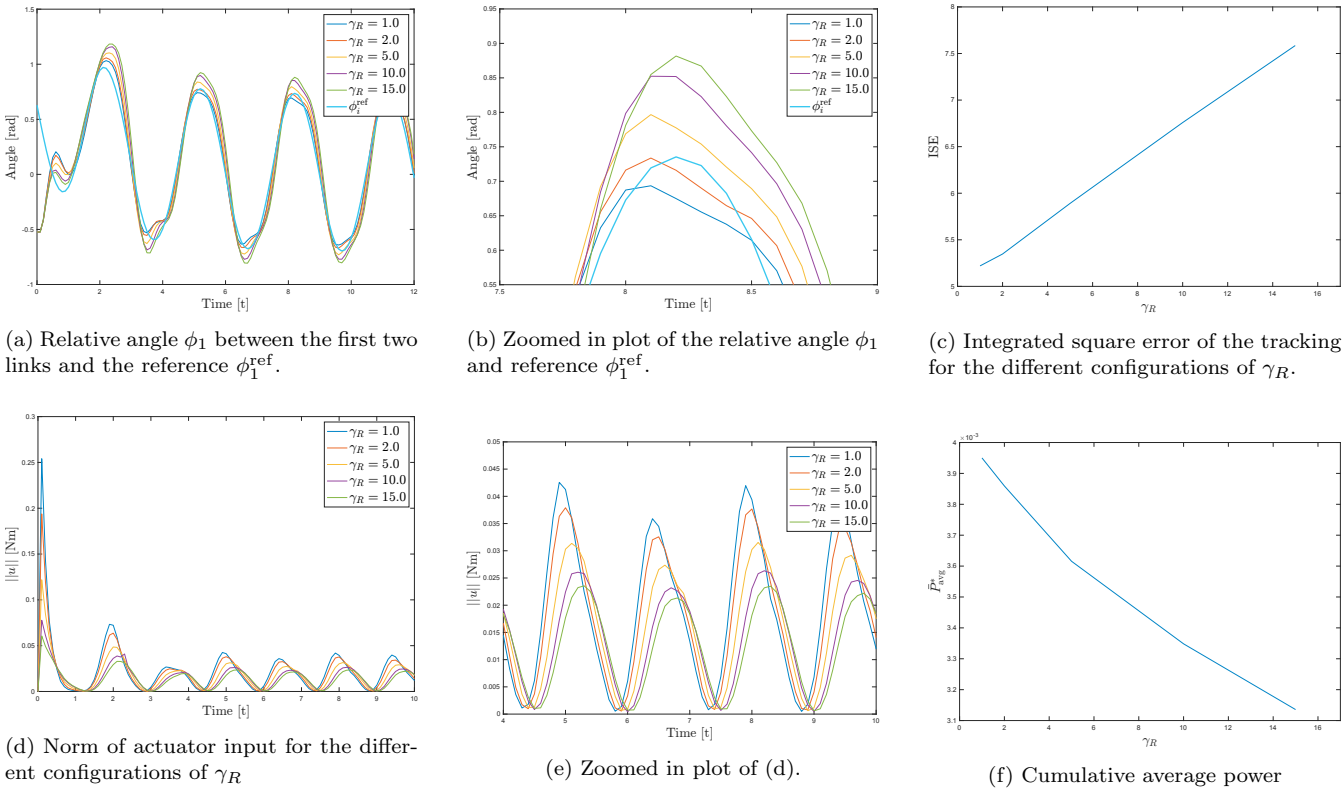
Future work involves investigating how the performance can be further improved by using more complex prediction models. By using prediction models that better capture the fluid-structure interaction, the swimming efficiency might be increased by utilizing the fluid-induced motion. Additionally, the robustness of the NMPC should be studied through environments with more disturbances, such as currents, waves, and turbulent streams.

REFERENCES

Andersson, J.A.E., Gillis, J., Horn, G., Rawlings, J.B., and Diehl, M. (2018). CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*.

Bernier, C., Gazzola, M., Chatelain, P., and Ronsse, R. (2018). Numerical simulations and development of drafting strategies for robotic swimmers at low reynolds number. In *Proc. IEEE International Conf. Biomedical Robotics and Biomechatronics*. Enschede, The Netherlands.

Bernier, C., Gazzola, M., Ronsse, R., and Chatelain, P. (2019). Simulations of propelling and energy harvesting



(a) Relative angle ϕ_1 between the first two links and the reference ϕ_1^{ref}.

(b) Zoomed in plot of the relative angle ϕ_1 and reference ϕ_1^{ref}.

(c) Integrated square error of the tracking for the different configurations of γ_R.

(d) Norm of actuator input for the different configurations of γ_R

(e) Zoomed in plot of (d).

(f) Cumulative average power

.

Fig. 5. Results from the high-fidelity simulation study with turning offset given by equation 18

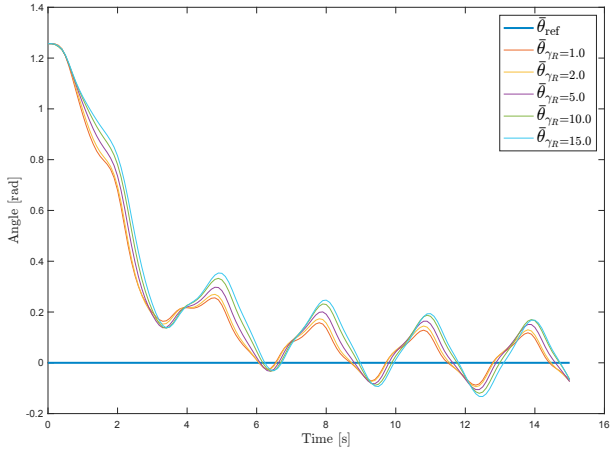


Fig. 6. Heading $\bar{\theta}$ and reference heading $\bar{\theta}_{\text{ref}}$.

articulated bodies via vortex particle-mesh methods. *J. of Computational Physics*, 392, 34–55.

Findeisen, R. and Allgöwer, F. (2002). An introduction to nonlinear model predictive control. In *Proc. 21st Benelux meeting on systems and control*, volume 11, 119–141. Veldhoven.

Fukushima, H., Yanagiya, T., Ota, Y., Katsumoto, M., and Matsuno, F. (2021). Model predictive path-following control of snake robots using an averaged model. *IEEE Transactions on Control Systems Technology*, 29(6), 2444–2456.

Guo, J. (2006). A waypoint-tracking controller for a biomimetic autonomous underwater vehicle. *Ocean Engineering*, 33(17), 2369–2380. doi: https://doi.org/10.1016/j.oceaneng.2005.11.012.

Kelasidi, E., Pettersen, K.Y., Gravdahl, J.T., and Liljebäck, P. (2014). Modeling of underwater snake robots. In *Proc. IEEE International Conf. on Robotics and Automation*.

Kelasidi, E., Liljebäck, P., Pettersen, K.Y., and Gravdahl, J.T. (2016). Innovation in underwater robots: Biologically inspired swimming snake robots. *IEEE Robotics & Automation Magazine*, 23(1), 44–62.

Liljebäck, P., Pettersen, K.Y., Stavdahl, Ø., and Gravdahl, J.T. (2013). *Snake Robots: Modelling, Mechatronics, and Control*. Advances in Industrial Control. Springer London.

Marafioti, G., Liljebäck, P., and Transeth, A.A. (2014). A study of nonlinear model predictive control for snake robot path following. In *Proc. 2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*, 568–573.

Nonhoff, M., Kohler, P.N., Kohl, A.M., Pettersen, K.Y., and Allgower, F. (2019). Economic model predictive control for snake robot locomotion. In *Proc. 2019 IEEE 58th Conference on Decision and Control (CDC)*, 8329–8334.

Rawlings, J.B., Mayne, D.Q., and Diehl, M. (2017). *Model predictive control: theory, computation, and design*, volume 2. Nob Hill Publishing Madison, WI.

Verschueren, R., Frison, G., Kouzoupis, D., Frey, J., van Duijkeren, N., Zanelli, A., Novoselnik, B., Albin, T., Quirynen, R., and Diehl, M. (2021). acados – a modular open-source framework for fast embedded optimal control. *Mathematical Programming Computation*.