Automatica 115 (2020) 108863

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Brief paper

Distributed implementation of nonlinear model predictive control for AUV trajectory tracking[☆]

Chao Shen, Yang Shi [*]

Department of Mechanical Engineering, University of Victoria, Victoria, BC, Canada, V8W 3P6

ARTICLE INFO

Article history:
Received 6 March 2019
Received in revised form 27 September 2019
Accepted 13 January 2020
Available online 14 February 2020

Keywords:
Autonomous underwater vehicle
Model predictive control
Trajectory tracking
Distributed control
Nonlinear control
Real-time control

ABSTRACT

This paper studies the trajectory tracking control of an autonomous underwater vehicle (AUV). We investigate the nonlinear model predictive control (NMPC) method looking for possible approaches to alleviate the heavy computational burden. Novel distributed NMPC algorithms are developed exploiting the dynamic properties of the AUV motion. By appropriately decomposing the original optimization problems into smaller size subproblems and then solving them in a distributed manner, the expected floating point operations (flops) can be reduced dramatically. We show that the convergence of the AUV trajectory can be guaranteed by the proposed contraction constraints in the decomposed subproblems. Recursive feasibility and closed-loop stability are proved. Taking advantage of the guaranteed stability, a real-time distributed implementation algorithm is further developed to automatically trade off between control performance and computational complexity. Extensive simulation studies on the Falcon AUV model demonstrate the effectiveness and robustness of the proposed approach.

1. Introduction

Trajectory tracking control is critical to AUV applications, but the design of reliable tracking controllers is challenging due to the nonlinearities in AUV dynamics and the uncertainties in the ocean environment (Li & Yan, 2017). In the past several decades, plenty of research effort has been devoted to this challenging topic (Shi, Shen, Fang, & Li, 2017). Traditional way-point tracking controllers can be designed using the classic Proportional–Integral–Derivative (PID), Linear–Quadratic–Regulator (LQR) and Kalman filtering techniques. Nonlinear PID (Lindegaard, 2003) can also be applied for possibly better control performance. When the desired trajectory is nonlinear, the curved trajectory emphasizes the nonlinearity in the vehicle motion and the classic linear control techniques are no longer applicable. Feedback linearization (Fossen, 2002) presents a powerful tool to deal with the nonlinearities. However, the main problem of applying feedback linearization to AUVs lies in the contradiction that it requires a high-fidelity system model, while the hydrodynamic coefficients for AUVs are rarely accurately identified. In view of this point, the Lyapunov-based backstepping control (BSC) becomes the mainstream design method for AUV tracking control. In Repoulias and Papadopoulos (2007) the kinematic controller was firstly designed and then the dynamic control law was derived using the BSC technique. An output BSC control for AUVs can be found in Do, Jiang, Pan, and Nijmeijer (2004). Another popular design approach for AUV trajectory tracking control is the sliding mode control (SMC) method well-known for its excellent robustness against parametric uncertainties. The side effect of SMC, however, seems to be the presence of “chattering” phenomenon. To alleviate the “chattering”, SMC controllers are usually used in combination with other control methods such as PID, robust control (Mahmoud, 2018; Soylu, Proctor, Podhorodeski, Bradley, & Buckham, 2015) and adaptive control (Soylu, Buckham, & Podhorodeski, 2008).

Nevertheless, a common deficiency of the preceding control methods appears to be the lacking capability of handling system constraints. System constraints are ubiquitous. For AUVs, they can be the physical limits of the actuators, the maximum allowed pitch angle for vessel stability, or the computing power of the onboard processor which are better be taken into consideration in the control system design. The model predictive control (MPC) is featured by the constraint handling capability (Mayne, 2014) which presents a powerful framework for solving a wide range of control problems. In addition, MPC is able to digest complex nonlinearity in the system dynamics making it an ideal candidate for solving AUV control problems. In terms of trajectory tracking control, there are only a few of pioneering investigations in the literature. In Naeem, Sutton, and Ahmad (2003) the linear MPC

[☆] This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC). The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Prashant Mhaskar under the direction of Editor Thomas Parisini.

[*] Corresponding author.
E-mail addresses: shenchao@uvic.ca (C. Shen), yshi@uvic.ca (Y. Shi).

https://doi.org/10.1016/j.automatica.2020.108863
0005-1098/© 2020 Elsevier Ltd. All rights reserved.

formulation was proposed. The generic algorithm was employed to solve the associated quadratic programming (QP) problem. The NMPC formulation for AUV trajectory tracking control was proposed in Guerreino, Silvestre, Cunha, and Pascoal (2014) where heuristic methods were developed to speed up the computation. In Shen, Shi, and Buckham (2017a), the integrated path planning and trajectory tracking control problem was studied. A unified receding horizon optimization framework was developed for AUVs to solve the combined motion control problem.

While MPC provides the benefit of fully incorporating nonlinear dynamic constraints, this benefit comes at price: a computational bottleneck is formed by the heavy burden of solving open-loop optimal control problems (OCPs) in real time. In theoretical studies, the computing time is usually omitted, but in engineering practice, the OCPs can only be solved by iterative methods. As far as nonlinear dynamics (e.g., AUV) are concerned, the OCPs present generic nonlinear programming (NLP) problems. The computational complexity raises exponentially as the problem size increases. Since the sampling period is practically short, many strategies, such as numerical continuation (Shen, Shi, & Buckham, 2017c), event-triggered control (Li & Shi, 2014), and off-line precomputation (Buskens & Maurer, 2000), have been proposed to reduce the computational complexity of MPC. In Shen, Shi, and Buckham (2016), by exploiting the dynamic properties of the AUV motion, a distributed computation scheme was successfully implemented for the NMPC tracking control. However, in Shen et al. (2016) the closed-loop stability was not proved. Since MPC calculates the control signal by solving OCPs, the control signal is implicitly related to the system state, which complicates the analysis of the closed-loop stability. Standard MPC design employs additional terminal constraints to the OCP and constructs an auxiliary stabilizing control law through local linearization (Mayne, 2014). For AUV tracking control, however, it is inappropriate to perform the local linearization due to the time-varying nature of the desired trajectory. A better solution needs to be found.

In this paper, we are looking to provide a way to eliminate the cost of implementing NMPC while guaranteeing the closed-loop stability at the same time. To this end, a reference augmentation technique is applied so that the coupling between surge, sway and yaw motions is weakened. Then the distributed optimization is exploited to alleviate the computation burden of applying the NMPC. Novel distributed NMPC algorithms are developed. By solving three subproblems with smaller size, the computational complexity can be reduced significantly. The main contributions of this paper are as follows:

- A well-defined NMPC formulation is provided for the AUV trajectory tracking control so that the dynamic properties of the AUV motion can be exploited.
- Three novel distributed NMPC algorithms are developed. The computational complexity of the tracking control can be dramatically reduced while the tracking performance can be well maintained.
- The key properties of the closed-loop system are analyzed explicitly. Sufficient conditions for recursive feasibility and closed-loop stability are provided.
- Extensive simulation studies reveal the prominent robustness of the proposed distributed NMPC.

The remainder of this paper is organized as follows: Section 2 describes the AUV model. In Section 3, the NMPC formulation is introduced and the computational complexity is briefed. Section 4 presents the proposed novel distributed NMPC algorithms along with theoretical results. In Section 5, the proposed algorithms are verified through extensive simulation studies. Section 6 concludes the paper.

2. System modeling

In this paper, we consider the horizontal motion of the AUV. The kinematic equations can be expressed in the following vectorial form (Fossen, 2002):

$$\dot{\boldsymbol{\eta}} = \mathbf{R}(\psi)\mathbf{v} \tag{1}$$

where $\boldsymbol{\eta} = [x, y, \psi]^{\mathrm{T}}$ is the position and heading vector; $\mathbf{v} = [u, v, r]^{\mathrm{T}}$ is the velocity vector; $\mathbf{R}(\psi)$ is the standard rotation matrix. Expanding (1) we have

$$\begin{aligned} \dot{x} &= u\cos\psi - v\sin\psi \\ \dot{y} &= u\sin\psi + v\cos\psi \\ \dot{\psi} &= r \end{aligned} \tag{2}$$

The dynamic equations analyze the cause of the motion, and can be established via laws of Newton:

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{C}(\mathbf{v})\mathbf{v} + \mathbf{D}(\mathbf{v})\mathbf{v} = \boldsymbol{\tau} + \mathbf{w} \tag{3}$$

where $\boldsymbol{\tau} = [F_u, F_v, F_r]^{\mathrm{T}}$ is the generalized thrust; $\mathbf{w}$ represents the external disturbances; $\mathbf{M}$ is the inertia matrix including added mass. $\mathbf{C}(\mathbf{v})$ denotes the Coriolis and centripetal matrix. The matrix $\mathbf{D}(\mathbf{v})$ captures the hydrodynamic damping mainly including drag force and vortex-induced force. Since MPC has prominent robustness (as will be seen in Section 5), in the controller design, we do not explicitly consider the external disturbances. Expanding the dynamic equations, we have

$$\dot{u} = \frac{M_{\dot{v}}}{M_{\dot{u}}}vr - \frac{X_u}{M_{\dot{u}}}u - \frac{D_u}{M_{\dot{u}}}u|u| + \frac{F_u}{M_{\dot{u}}} \tag{4a}$$

$$\dot{v} = -\frac{M_{\dot{u}}}{M_{\dot{v}}}ur - \frac{Y_v}{M_{\dot{v}}}v - \frac{D_v}{M_{\dot{v}}}v|v| + \frac{F_v}{M_{\dot{v}}} \tag{4b}$$

$$\dot{r} = \frac{M_{\dot{u}} - M_{\dot{v}}}{M_{\dot{r}}}uv - \frac{N_r}{M_{\dot{r}}}r - \frac{D_r}{M_{\dot{r}}}r|r| + \frac{F_r}{M_{\dot{r}}} \tag{4c}$$

The thrust allocation can be described by $\boldsymbol{\tau} = \mathbf{B}\mathbf{u}$ where $\mathbf{u}$ denotes the force provided by each thruster; $\mathbf{B}$ is the input matrix. Then the model used for the AUV trajectory tracking control design is established as follows:

$$\dot{\mathbf{x}} = \begin{bmatrix} \mathbf{R}(\psi)\mathbf{v} \\ \mathbf{M}^{-1}(\mathbf{B}\mathbf{u} - \mathbf{C}\mathbf{v} - \mathbf{D}\mathbf{v}) \end{bmatrix} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \tag{5}$$

where $\mathbf{x} = [\boldsymbol{\eta}^{\mathrm{T}}, \mathbf{v}^{\mathrm{T}}]^{\mathrm{T}}$ denotes the motion state and $\mathbf{u} = [u_1, u_2, u_3, u_4]^{\mathrm{T}}$ lists the forces the thrusters provide.

3. NMPC tracking control

The desired trajectory is usually defined in the output space: $p(t) = [x_d(t), y_d(t)]^{\mathrm{T}}$ which describes the desired position in a global reference frame.

Assumption 1. The desired trajectory $p(t)$ and its derivatives are bounded satisfying: $0 \le \underline{p} \le \|p(t)\|_\infty \le \bar{p} < \infty$, $0 < \underline{p_1} \le \|\dot{p}(t)\|_\infty \le \bar{p}_1 < \infty$, $0 \le \underline{p_2} \le \|\ddot{p}(t)\|_\infty \le \bar{p}_2 < \infty$ and $0 \le \underline{p_3} \le \|\dddot{p}(t)\|_\infty \le \bar{p}_3 < \infty$.

With the desired trajectory $p(t)$ we do the following reference augmentation: let $\mathbf{x}_d(t) = [x_d(t), y_d(t), \psi_d(t), u_d(t), v_d(t), r_d(t)]^{\mathrm{T}}$ with

$$\begin{aligned} &\psi_d(t) = \operatorname{atan2}(\dot{y}_d, \dot{x}_d), \ u_d(t) = \sqrt{\dot{x}_d^2 + \dot{y}_d^2} \\ &v_d(t) = 0, \ r_d(t) = (\dot{x}_d\ddot{y}_d - \dot{y}_d\ddot{x}_d)/(\dot{x}_d^2 + \dot{y}_d^2) \end{aligned} \tag{6}$$

where atan2 is the four-quadrant inverse tangent operator. It is easy to verify that $\mathbf{x}_d(t)$ satisfies the kinematic equations of motion, i.e., $\dot{\boldsymbol{\eta}}_d = \mathbf{R}(\psi_d)\mathbf{v}_d$. Therefore, each state of the AUV motion now has a unique reference.

Then the optimal control problem for the AUV tracking can be formulated as follows,

$$\min_{\mathbf{u}\in S(\delta)} J = \int_0^T \|\tilde{\mathbf{x}}(s)\|_Q^2 + \|\mathbf{u}(s)\|_R^2 ds + \|\tilde{\mathbf{x}}(T)\|_P^2 \tag{7a}$$

$$\text{s.t.} \quad \dot{\mathbf{x}}(s) = \mathbf{f}(\mathbf{x}(s), \mathbf{u}(s)) \tag{7b}$$

$$\mathbf{x}(0) = \mathbf{x}(t_0), \ \ |\mathbf{u}(s)| \le \mathbf{u}_{\max} \tag{7c}$$

where $\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{x}_d$ denotes the error state; $S(\delta)$ is the family of piecewise constant functions characterized by the sampling period δ; $T = N\delta$ is known as the prediction horizon, and $Q = \text{diag}\{q_{ii}\}$, $R = \text{diag}\{r_{ii}\}$, $P = \text{diag}\{p_{ii}\}$ are weighting matrices, positive definite.

In (7), the performance index J includes integral operations and the constraints encompass derivative operations. Both the integral and derivative operations are performed numerically and the following optimization problem is solved at each sampling time:

$$\min_U J = \sum_{k=0}^{N-1} (\|\tilde{\mathbf{x}}(k)\|_Q^2 + \|\mathbf{u}(k)\|_R^2) + \|\tilde{\mathbf{x}}(N)\|_P^2 \tag{8a}$$

$$\text{s.t.} \quad \mathbf{x}(k+1) = f(\mathbf{x}(k), \mathbf{u}(k)) \tag{8b}$$

$$\mathbf{x}(0) = \mathbf{x}(t_0), \ \ |\mathbf{u}(k)| \le \mathbf{u}_{\max} \tag{8c}$$

Here, the system model is discretized and represented by $\mathbf{x}(k+1) = f(\mathbf{x}(k), \mathbf{u}(k))$ with $\mathbf{x}(i) = \mathbf{x}(i\delta)$ and $\mathbf{u}(i) = \mathbf{u}(i\delta)$; the sequence of control inputs $U = [\mathbf{u}^{\mathrm{T}}(0), \mathbf{u}^{\mathrm{T}}(1), \ldots, \mathbf{u}^{\mathrm{T}}(N-1)]^{\mathrm{T}}$ are the decision variables of the optimization problem. The discretization is assumed to be stable and it provides enough accuracy.

Since f is non-convex and sits in the constraints, solving (8) is a generic NLP problem. The computational complexity of solving (8) using the sequential quadratic programming (SQP) method can be estimated by expected flops. The worst-case number of flops for each QP subproblem can be approximated by Richter, Jones, and Morari (2012)

$$\#\text{flops}_{\text{IP}} = i_{\text{IP}}(2/3(Nm)^3 + 2(Nm)^2) \tag{9}$$

where N is the prediction horizon and m is the number of control inputs, hence Nm is the number of decision variables for the optimization problem; i_{IP} is the number of interior point iterations which is expected to be $\mathcal{O}(\sqrt{Nm}\log(1/\epsilon))$ (McGovern, 2000) for the ϵ-accurate solution. Neglecting minor operations in function evaluations, gradient and Hessian updating, and line search in one major SQP iteration, a rough approximation of the computational complexity for solving (8) is

$$\#\text{flops}_{\text{SQP}} \approx i_{\text{SQP}} \times (\#\text{flops}_{\text{IP}}) \tag{10}$$

where i_{SQP} denotes the numbers of SQP iterations which is specified by the maximum iteration number $i_{\text{SQP,max}}$. From (10) we know that shrinking m and N effectively reduces the computational complexity.

4. Distributed NMPC implementation

In this section, we propose a distributed optimization strategy for implementing the NMPC tracking control such that the complexity can be reduced dramatically.

4.1. Distributed implementation

To facilitate the distributed implementation, we decouple the thrust allocation (TA) from the tracking control law calculation: Provided that $\mathbf{B}^{\mathrm{T}}\mathbf{B}$ is nonsingular, the Moore–Penrose pseudoinverse method can be adopted as the closed-form solution for the TA:

$$\mathbf{u} = (\mathbf{B}^{\mathrm{T}}\mathbf{B})^{-1}\mathbf{B}^{\mathrm{T}}\boldsymbol{\tau} = \mathbf{B}^{+}\boldsymbol{\tau} \tag{11}$$

Then we choose $\boldsymbol{\tau} = [F_u, F_v, F_r]^{\mathrm{T}}$ as the decision variables in (7). Furthermore, instead of using $|\mathbf{B}^{+}\boldsymbol{\tau}(s)| \le \mathbf{u}_{\max}$ in place of (7c), we use $|\boldsymbol{\tau}(s)| \le \boldsymbol{\tau}_{\max}$ which sets a direct bound constraint on the decision variables.

Assumption 2. The thrusters have the same maximum capacity, i.e., $|u_i| \le u_{\max}$.

Then we have the following proposition to guide the choice of $\boldsymbol{\tau}_{\max}$.

Proposition 1 (Shen, Shi, & Buckham, 2017b)**.** *Consider the TA using the closed-form solution (11). Specify the maximum generalized thrust by $\tau_{max} = \|\boldsymbol{\tau}_{max}\|_\infty$ with $\boldsymbol{\tau}_{max} = [F_{u,max}, F_{v,max}, F_{r,max}]^{\mathrm{T}}$. If the following can hold,*

$$\tau_{max} \le \frac{u_{max}}{\bar{b}^{+}} \tag{12}$$

where $\bar{b}^{+} = \|\mathbf{B}^{+}\|_\infty$, then the TA is always feasible for the real system, i.e., $\|\mathbf{u}\|_\infty \le u_{max}$.

Remark 1. Although choosing $\tau_{\max}$ that satisfies (12) guarantees the TA always fits the real system, it introduces some degree of conservativeness: Using $|\boldsymbol{\tau}(s)| \le \boldsymbol{\tau}_{\max}$ tightens the constraints since (12) is only a sufficient condition rather than the sufficient and necessary condition. However, the constraints tightening is important for the following distributed implementation because the original constraints $|\mathbf{B}^{+}\boldsymbol{\tau}(s)| \le \mathbf{u}_{\max}$ couple the decision variables. With the tightened constraints the decision variables $\boldsymbol{\tau} = [F_u, F_v, F_r]^{\mathrm{T}}$ are independent.

From (4) and (6) we find that along the reference trajectory the dynamic equations of motion are loosely coupled. In fact, the surge dynamics (4a) and yaw dynamics (4c) are totally decoupled since $v = 0$. This observation inspires the idea of distributed implementation for approximately solving the NMPC problem.

In the distributed control paradigm, the tracking control signals are calculated separately by three subsystems. The surge subsystem only considers the surge dynamics and the kinematics, and determines the control input F_u by solving the following subproblem:

$$\min_{F_u\in S(\delta)} J_1 = \int_0^T \|\tilde{\mathbf{x}}(s)\|_Q^2 + r_{11}F_u^2(s) ds + \|\tilde{\mathbf{x}}(T)\|_P^2 \tag{13a}$$

$$\text{s.t.} \quad \dot{\xi}(s) = f_1(\xi(s), \hat{v}(s), \hat{r}(s), F_u(s)) \tag{13b}$$

$$\xi(0) = \xi(t_0), \ \ |F_u(s)| \le F_{\text{u,max}} \tag{13c}$$

where $\xi = [x, y, \psi, u]^{\mathrm{T}}$ is the state of the surge subsystem and the subsystem model f_1 can be elaborated by taking the corresponding columns and rows in (5). The f_1 also contains v and r but they can be viewed as known coefficients for some assumed trajectories $\hat{v}(s)$ and $\hat{r}(s)$.

Similarly, the sway subsystem includes the sway dynamics and the kinematics, and determines the control input F_v by solving the subproblem:

$$\min_{F_v\in S(\delta)} J_2 = \int_0^T \|\tilde{\mathbf{x}}(s)\|_Q^2 + r_{22}F_v^2(s) ds + \|\tilde{\mathbf{x}}(T)\|_P^2 \tag{14a}$$

$$\text{s.t.} \quad \dot{\zeta}(s) = f_2(\zeta(s), \hat{u}(s), \hat{r}(s), F_v(s)) \tag{14b}$$

$$\zeta(0) = \zeta(t_0), \ \ |F_v(s)| \le F_{\text{v,max}} \tag{14c}$$

where $\zeta = [x, y, \psi, v]^{\mathrm{T}}$ is the state of the sway subsystem and the model f_2 can be obtained by taking the corresponding columns and rows in (5).

The yaw subsystem encompasses the yaw dynamics and the kinematics, and determines the control input F_r by solving the following optimization problem:

$$\min_{F_r \in S(\delta)} J_3 = \int_0^T \|\tilde{\mathbf{x}}(s)\|_Q^2 + r_{33}F_r^2(s)ds + \|\tilde{\mathbf{x}}(T)\|_P^2 \tag{15a}$$

$$\text{s.t.} \quad \dot{\omega}(s) = f_3(\omega(s), \hat{u}(s), \hat{v}(s), F_r(s)) \tag{15b}$$

$$\omega(0) = \omega(t_0), \ \ |F_r(s)| \leq F_{\text{r,max}} \tag{15c}$$

where $\omega = [x, y, \psi, r]^{\text{T}}$ is the state of the yaw subsystem and the dynamics f_3 can be explicated by picking out the corresponding columns and rows in (5).

The assumed state trajectories $\hat{u}$, $\hat{v}$ and $\hat{r}$ can be determined in the following way: Let $\mathbf{F}_u^* = [F_u^*(0), \ldots, F_u^*(N-1)]^{\text{T}}$, $\mathbf{F}_v^* = [F_v^*(0), \ldots, F_v^*(N-1)]^{\text{T}}$, $\mathbf{F}_r^* = [F_r^*(0), \ldots, F_r^*(N-1)]^{\text{T}}$ be the optimal solutions of the subproblems (13), (14) and (15) for the previous time. At the current time, construct the assumed control signals using the previous solution as

$$\hat{\mathbf{F}}_u = [F_u^*(1), \ldots, F_u^*(N-1), F_u^*(N-1)]^{\text{T}} \tag{16a}$$

$$\hat{\mathbf{F}}_v = [F_v^*(1), \ldots, F_v^*(N-1), F_v^*(N-1)]^{\text{T}} \tag{16b}$$

$$\hat{\mathbf{F}}_r = [F_r^*(1), \ldots, F_r^*(N-1), F_r^*(N-1)]^{\text{T}} \tag{16c}$$

Then the assumed state trajectories $\hat{u}$, $\hat{v}$ and $\hat{r}$ can be obtained via state evolution through the dynamic equations (4) and kinematic equations (2) starting from the updated system state $\mathbf{x}(t)$ for the current time.

An initialization procedure is included to facilitate the warm start: At the very first sampling instant, solve the centralized problem (7) once to obtain the assumed control trajectories $\hat{\boldsymbol{\tau}}(s) = \boldsymbol{\tau}^*(s)$ for $t = 0$. The distributed control algorithm is summarized in Algorithm 1. By solving the subproblems in parallel, the algorithmic efficiency can be improved dramatically.

Algorithm 1 (Distributed Implementation)

```
1: Initialization: Input the weighting matrices and prediction horizon; solve (7) at t=0, let $\mathbf{u}^*(s)$ denote the solution; set $\hat{\boldsymbol{\tau}}(s) = \boldsymbol{\tau}^*(s) = \mathbf{B}\mathbf{u}^*(s)$.
2: Fetch the state measurement $\mathbf{x}(t)$.
3: Calculate the assumed state trajectories by $\dot{\hat{\mathbf{x}}} = \mathbf{f}(\hat{\mathbf{x}}, \mathbf{B}^+\hat{\boldsymbol{\tau}})$ with $\hat{\mathbf{x}}(0) = \mathbf{x}(t)$.
4: Solve the subproblems (13)–(15) in parallel; let $\boldsymbol{\tau}^*(s) = [F_u^*(s), F_v^*(s), F_r^*(s)]^{\text{T}}$ denote the solution.
5: Set $\mathbf{u}^*(s) = \mathbf{B}^+\boldsymbol{\tau}^*(s)$; construct $\hat{\boldsymbol{\tau}}(s)$ using (16).
6: Implement $\mathbf{u}^*(s)$ for only one sampling period: $\mathbf{u}(t) = \mathbf{u}^*(s)$ for $s \in [0, \delta]$.
7: At next sampling time, set $t = t + \delta$; Goto step 2.
```

4.2. Sacrificing efficiency for stability

So far, we have only focused on the computational complexity, and we had no explicit consideration of the closed-loop stability. The optimality of the solutions does not indicate the stability for the distributed implementation in Algorithm 1, not even for the centralized NMPC. To theoretically ensure the closed-loop stability, we modify the subproblems and solve them sequentially.

Assume that $\mathbf{v}_d = [u_d, v_d, r_d]^{\text{T}}$ satisfies the dynamic equations of AUV motion, then we can have the reference control forces $\boldsymbol{\tau}_d = [F_{ud}, F_{vd}, F_{rd}]^{\text{T}}$ calculated by

$$\boldsymbol{\tau}_d = \mathbf{M}\dot{\mathbf{v}}_d + \mathbf{C}(\mathbf{v}_d)\mathbf{v}_d + \mathbf{D}(\mathbf{v}_d)\mathbf{v}_d \tag{17}$$

where $\dot{\mathbf{v}}_d$ is obtained by taking the time derivative of (6).

Define $\tilde{F}_u = F_u - F_{ud}$, $\tilde{F}_v = F_v - F_{vd}$ and $\tilde{F}_r = F_r - F_{rd}$ and modify the surge subproblem as follows:

$$\min_{F_u \in S(\delta)} J_1 = \int_0^T \|\tilde{\mathbf{x}}(s)\|_Q^2 + r_{11}\tilde{F}_u^2(s)ds + \|\tilde{\mathbf{x}}(T)\|_P^2 \tag{18a}$$

$$\text{s.t.} \quad \dot{\xi}(s) = f_1(\xi(s), \hat{v}(s), \hat{r}(s), F_u(s)) \tag{18b}$$

$$\xi(0) = \xi(t_0), \ \ |F_u(s)| \leq F_{\text{u,max}} \tag{18c}$$

$$\tfrac{\partial V}{\partial \mathbf{x}}\mathbf{f}(\mathbf{x}(0), \lambda(0)) \leq \tfrac{\partial V}{\partial \mathbf{x}}\mathbf{f}(\mathbf{x}(0), h(\mathbf{x}(0))) \tag{18d}$$

where $h(\mathbf{x}) = \mathbf{B}^+\bar{h}(\mathbf{x})$ with $\bar{h}(\mathbf{x}) = [\bar{h}_u(\mathbf{x}), \bar{h}_v(\mathbf{x}), \bar{h}_r(\mathbf{x})]^T$ is an auxiliary tracking controller and $V(\mathbf{x})$ is the corresponding Lyapunov function; $\lambda(0) = \mathbf{B}^+\bar{\lambda}(0)$ with $\bar{\lambda}(0) = [F_u(0), \bar{h}_v(\mathbf{x}(0)), \bar{h}_r(\mathbf{x}(0))]^T$.

The modified subproblem for the sway subsystem is constructed as follows:

$$\min_{F_v \in S(\delta)} J_2 = \int_0^T \|\tilde{\mathbf{x}}(s)\|_Q^2 + r_{22}\tilde{F}_v^2(s)ds + \|\tilde{\mathbf{x}}(T)\|_P^2 \tag{19a}$$

$$\text{s.t.} \quad \dot{\zeta}(s) = f_2(\zeta(s), \hat{u}(s), \hat{r}(s), F_v(s)) \tag{19b}$$

$$\zeta(0) = \zeta(t_0), \ \ |F_v(s)| \leq F_{\text{v,max}} \tag{19c}$$

$$\tfrac{\partial V}{\partial \mathbf{x}}\mathbf{f}(\mathbf{x}(0), \pi(0)) \leq \tfrac{\partial V}{\partial \mathbf{x}}\mathbf{f}(\mathbf{x}(0), h(\mathbf{x}(0))) \tag{19d}$$

where $\pi(0) = \mathbf{B}^+\bar{\pi}(0)$ and $\bar{\pi}(0) = [F_u^*(0), F_v(0), \bar{h}_r(\mathbf{x}(0))]^T$; $F_u^*(s)$ is the solution of (18) passed from the surge subsystem. And the optimization problem for the yaw subsystem is modified as follows:

$$\min_{F_r \in S(\delta)} J_3 = \int_0^T \|\tilde{\mathbf{x}}(s)\|_Q^2 + r_{33}\tilde{F}_r^2(s)ds + \|\tilde{\mathbf{x}}(T)\|_P^2 \tag{20a}$$

$$\text{s.t.} \quad \dot{\omega}(s) = f_3(\omega(s), \hat{u}(s), \hat{v}(s), F_r(s)) \tag{20b}$$

$$\omega(0) = \omega(t_0), \ \ |F_r(s)| \leq F_{\text{r,max}} \tag{20c}$$

$$\tfrac{\partial V}{\partial \mathbf{x}}\mathbf{f}(\mathbf{x}(0), \mathbf{u}(0)) \leq \tfrac{\partial V}{\partial \mathbf{x}}\mathbf{f}(\mathbf{x}(0), h(\mathbf{x}(0))) \tag{20d}$$

where $\mathbf{u}(0) = \mathbf{B}^+\boldsymbol{\tau}(0)$ and $\boldsymbol{\tau}(0) = [F_u^*(0), F_v^*(0), F_r(0)]^T$; and $F_u^*(s)$, $F_v^*(s)$ are the solutions of (18) and (19).

The modified distributed control algorithm is summarized in Algorithm 2, and we have the following feasibility and stability theorems.

Algorithm 2 (Modified Distributed Implementation)

```
1: Initialization: Input the weighting matrices and prediction horizon; solve (7) at t=0, let $\mathbf{u}^*(s)$ denote the solution; set $\hat{\boldsymbol{\tau}}(s) = \boldsymbol{\tau}^*(s) = \mathbf{B}\mathbf{u}^*(s)$.
2: Fetch the state measurement $\mathbf{x}(t)$.
3: Calculate the assumed state trajectories by $\dot{\hat{\mathbf{x}}} = \mathbf{f}(\hat{\mathbf{x}}, \mathbf{B}^+\hat{\boldsymbol{\tau}})$ with $\hat{\mathbf{x}}(0) = \mathbf{x}(t)$.
4: Solve (18) and send the solution $F_u^*(s)$ to the sway and yaw subsystems.
5: Solve (19) and send $F_v^*(s)$ to the yaw subsystem.
6: Solve (20) and let the solution be $F_r^*(s)$.
7: Set $\mathbf{u}^*(s) = \mathbf{B}^+\boldsymbol{\tau}^*(s)$ with $\boldsymbol{\tau}^*(s) = [F_u^*(s), F_v^*(s), F_r^*(s)]^{\text{T}}$; construct $\hat{\boldsymbol{\tau}}(s)$ using (16).
8: Implement $\mathbf{u}^*(s)$ for one sampling period: $\mathbf{u}(t) = \mathbf{u}^*(s)$ for $s \in [0, \delta]$. Set $t = t + \delta$; Goto step 2.
```

Theorem 1. *Choosing $F_{u,max} = F_{v,max} = F_{r,max} = \tau_{max}$ which satisfies (12), if $\|\bar{h}(\mathbf{x})\|_\infty \leq \tau_{max}$ can hold, then the Algorithm 2 admits recursively feasibility, i.e., we can always find an initial feasible solution for each of the optimizations (18)–(20) to start with.*

Proof. As $\|\bar{h}(\mathbf{x})\|_\infty \leq \tau_{\text{max}}$, we have $|\bar{h}_u(\mathbf{x})| \leq F_{\text{u,max}}$. Then the $\bar{h}_u(\mathbf{x})$ can be used as a feasible initial solution for the surge subproblem (18).

For the sway subsystem, since $|\bar{h}_v(\mathbf{x})| \leq F_{\text{v,max}}$ and $F_u^*(s)$ satisfies (18d), it can be easily verified that $\bar{h}_v(\mathbf{x})$ must be a feasible solution for the subproblem (19).

Likewise, we have $|\bar{h}_r(\mathbf{x})| \leq F_{\text{r,max}}$ and $F_u^*(s)$, $F_v^*(s)$ satisfying (19d). Therefore, $\bar{h}_r(\mathbf{x})$ is feasible for (20). ■

Theorem 2. *Suppose that there exists an auxiliary control law $\bar{h}(\mathbf{x})$ such that $\tilde{\mathbf{x}} = \mathbf{0}$ is asymptotically stable for the closed-loop system controlled by $h(\mathbf{x}) = \mathbf{B}^{+}\bar{h}(\mathbf{x})$; $V(\mathbf{x})$ is the corresponding Lyapunov function. Provided that the recursive feasibility can be guaranteed, then the closed-loop system under Algorithm 2 is asymptotically stable, i.e., the AUV converges to the desired trajectory.*

Proof. By converse Lyapunov theorems (Khalil, 1996), there exist some functions $\gamma_i(\cdot), i = 1, 2, 3$ which belong to the class $\mathcal{K}_\infty$ such that the following inequalities hold:

$$\gamma_1(\|\mathbf{x}\|) \le V(\mathbf{x}) \le \gamma_2(\|\mathbf{x}\|) \tag{21a}$$

$$\frac{\partial V}{\partial \mathbf{x}}\mathbf{f}(\mathbf{x}, h(\mathbf{x})) \le -\gamma_3(\|\mathbf{x}\|) \tag{21b}$$

Since we have the contraction constraint (20d) and $\mathbf{u}(t) = \mathbf{u}^*(s)$ only for $s \in [0, \delta]$, the following holds:

$$\frac{\partial V}{\partial \mathbf{x}}\mathbf{f}(\mathbf{x}, \mathbf{u}(\mathbf{x})) \le \frac{\partial V}{\partial \mathbf{x}}\mathbf{f}(\mathbf{x}, h(\mathbf{x})) \le -\beta_3(\|\mathbf{x}\|) \tag{22}$$

Then by standard Lyapunov arguments (Theorem 4.8 in Khalil, 1996) we claim that $\tilde{\mathbf{x}} = \mathbf{0}$ is asymptotically stable for the closed-loop system controlled by Algorithm 2, i.e., the AUV is guaranteed to converge to the desired trajectory. ■

Note that Theorems 1 and 2 depend on the auxiliary control law $\bar{h}(\mathbf{x})$. In principal, the construction of $\bar{h}(\mathbf{x})$ can be done by any Lyapunov-based control design. Here, we provide an example with the nonlinear backstepping control law.

Following the standard backstepping control design procedure for AUV tracking control (Fossen, 2002), we can derive the auxiliary control law $\bar{h}(\mathbf{x})$ as follows:

$$\bar{h}(\mathbf{x}) = \mathbf{M}\dot{\mathbf{v}}_r + \mathbf{C}\mathbf{v}_r + \mathbf{D}\mathbf{v}_r - \mathbf{R}^{\mathrm{T}}\mathbf{K}_p\tilde{\boldsymbol{\eta}} - \mathbf{R}^{\mathrm{T}}\mathbf{K}_d\mathbf{s} \tag{23}$$

where $\mathbf{K}_p$ and $\mathbf{K}_d$ are control gains, positive definite; $\tilde{\boldsymbol{\eta}} = \boldsymbol{\eta} - \boldsymbol{\eta}_d$, $\mathbf{v}_r = \mathbf{R}^{\mathrm{T}}(\psi)\dot{\boldsymbol{\eta}}_r$, $\dot{\boldsymbol{\eta}}_r = \dot{\boldsymbol{\eta}}_d - \tilde{\boldsymbol{\eta}}$ and $\mathbf{s} = \dot{\boldsymbol{\eta}} - \dot{\boldsymbol{\eta}}_r$. The corresponding Lyapunov function $V(\mathbf{x})$ is

$$V(\mathbf{x}) = \frac{1}{2}\mathbf{s}^{\mathrm{T}}\mathbf{M}^*(\psi)\mathbf{s} + \frac{1}{2}\tilde{\boldsymbol{\eta}}^{\mathrm{T}}\mathbf{K}_p\tilde{\boldsymbol{\eta}} \tag{24}$$

where $\mathbf{M}^*(\psi) = \mathbf{R}(\psi)\mathbf{M}\mathbf{R}^{\mathrm{T}}(\psi)$. Then the detailed expression of the surge subproblem contraction constraint (18d) corresponding to (23) is

$$\begin{aligned} &-\mathbf{s}(0)^{\mathrm{T}}\mathbf{D}^*(\mathbf{v}(0), \psi(0))\mathbf{s}(0) + \mathbf{s}(0)^{\mathrm{T}}\mathbf{R}(\psi(0))[\lambda(0) \\ &\quad - \mathbf{M}\dot{\mathbf{v}}_r(0) - \mathbf{C}(\mathbf{v}(0))\mathbf{v}_r(0) - \mathbf{D}(\mathbf{v}(0))\mathbf{v}_r(0) \\ &\quad - \mathbf{g}(\boldsymbol{\eta}(0))] - \tilde{\boldsymbol{\eta}}(0)^{\mathrm{T}}\mathbf{K}_p\tilde{\boldsymbol{\eta}}(0) + \mathbf{s}(0)^{\mathrm{T}}\mathbf{K}_p\tilde{\boldsymbol{\eta}}(0) \\ &\le -\mathbf{s}(0)^{\mathrm{T}}[\mathbf{D}^*(\mathbf{v}(0), \psi(0)) + \mathbf{K}_d]\mathbf{s}(0) - \tilde{\boldsymbol{\eta}}(0)^{\mathrm{T}}\mathbf{K}_p\tilde{\boldsymbol{\eta}}(0) \end{aligned} \tag{25}$$

where $\mathbf{D}(\mathbf{v}, \psi)^* = \mathbf{R}(\psi)\mathbf{D}(\mathbf{v})\mathbf{R}^{\mathrm{T}}(\psi)$. The contraction constraints for sway subproblem (19d) and for yaw subproblem (20d) are in similar forms.

Remember that the auxiliary control law must satisfy $\|\bar{h}(\mathbf{x})\|_\infty \le \tau_{\max}$ to guarantee the recursive feasibility of Algorithm 2 and the stability of the closed-loop system. We have the following theoretical results:

Theorem 3 (*Shen, Shi, & Buckham, 2018*). *Given control gain matrices $\mathbf{K}_p = \mathrm{diag}\{k_{pi}\}$ and $\mathbf{K}_d = \mathrm{diag}\{k_{di}\}$, positive definite. Denote $\bar{k}_p = \max\{k_{pi}\}$, $\bar{k}_d = \max\{k_{di}\}$, $\bar{m} = \max\{M_{\dot{u}}, M_{\dot{v}}, M_{\dot{r}}\}$, $\bar{d}_1 = \max\{X_u, Y_v, N_r\}$ and $\bar{d}_2 = \max\{D_u, D_v, D_r\}$, and define $\chi(t) = [\tilde{\boldsymbol{\eta}}^{\mathrm{T}}(t), \mathbf{s}^{\mathrm{T}}(t)]^{\mathrm{T}}$. If the following condition can be satisfied*

$$\bar{m}\bar{v}_{r1} + (\bar{c} + \bar{d})\bar{v}_r + \sqrt{2}(\bar{k}_p + \bar{k}_d)\|\chi(0)\|_2 \le \tau_{max} \tag{26}$$

where $\bar{c} = 2\sqrt{2}\bar{m}\bar{\eta}_{d1} + 4\sqrt{2}\bar{m}\|\chi(0)\|_2$, $\bar{d} = \bar{d}_1 + \bar{d}_2(\sqrt{2}\bar{\eta}_{d1} + 2\sqrt{2}\|\chi(0)\|_2)$, $\bar{v}_{r1} = \sqrt{2}\bar{\eta}_{d2} + 2\bar{\eta}_{d1}^2 + (2\sqrt{2} + 6\bar{\eta}_{d1})\|\chi(0)\|_2 + 4\|\chi(0)\|_2^2$ and $\bar{v}_r = \sqrt{2}(\bar{\eta}_{d1} + \|\chi(0)\|_2)$, then $\|\bar{h}(\mathbf{x})\|_\infty \le \tau_{max}$.

4.3. Negotiating between performance and complexity

Note that Algorithm 2 is not as efficient as Algorithm 1. However, since the closed-loop stability is always guaranteed regardless of the quality of optimal solution, we can adjust the maximum iteration number $i_{\mathrm{SQP,max}}$ (and the prediction horizon N) to achieve the real-time control. Given that the processing time t_{proc} can be fed back in real time, this adjustment can be performed automatically. A real-time DMPC algorithm is developed as Algorithm 3.

Algorithm 3 (Real-Time Distributed Implementation)

```
1: Initialization: Input the weighting matrices, prediction horizon $N$, maximum allowed processing time $t_{\max}$, and maximum iteration number $i_{\mathrm{SQP,max}}$; solve the NMPC problem (7) at t=0, let $\mathbf{u}^*(s)$ denote the solution; set $\hat{\boldsymbol{\tau}}(s) = \boldsymbol{\tau}^*(s) = \mathbf{B}\mathbf{u}^*(s)$.
2: Fetch the state measurement $\mathbf{x}(t)$.
3: Calculate the assumed state trajectories by $\dot{\hat{\mathbf{x}}} = \mathbf{f}(\hat{\mathbf{x}}, \mathbf{B}^{+}\hat{\boldsymbol{\tau}})$ with $\hat{\mathbf{x}}(0) = \mathbf{x}(t)$.
4: Solve (18) and send the solution $F_u^*(s)$ to the sway and yaw subsystems.
5: Solve (19) and send $F_v^*(s)$ to the yaw subsystem.
6: Solve (20) and let the solution be $F_r^*(s)$.
7: Set $\mathbf{u}^*(s) = \mathbf{B}^{+}\boldsymbol{\tau}^*(s)$ with $\boldsymbol{\tau}^*(s) = [F_u^*(s), F_v^*(s), F_r^*(s)]^{\mathrm{T}}$; construct $\hat{\boldsymbol{\tau}}(s)$ using (16).
8: Feed back the processing time $t_{\mathrm{proc}}$. If $t_{\mathrm{proc}} > t_{\max}$ then set $i_{\mathrm{SQP,max}} = i_{\mathrm{SQP,max}} - \Delta i_1$; else if $t_{\mathrm{proc}} < \alpha t_{\max}$ then set $i_{\mathrm{SQP,max}} = i_{\mathrm{SQP,max}} + \Delta i_2$.
9: Implement $\mathbf{u}^*(s)$ for one sampling period: $\mathbf{u}(t) = \mathbf{u}^*(s)$ for $s \in [0, \delta]$. Set $t = t + \delta$; Goto step 2.
```

In Algorithm 3 STEP 8, we use $\alpha \in [0, 1)$, Δi_1 and Δi_2 to control the increment and decrement of the maximum iteration number. These numbers need to be well tuned. Furthermore, the automatic adjustment of the prediction horizon N can also be incorporated.

Since modern optimization algorithms (such as primal–dual interior point methods, SQP) ensure constraint satisfaction and cost function improvement for each iteration, the Algorithm 3 essentially trades off between control performance and computational complexity in accordance with the current available onboard resources. Therefore, it is promising for low-cost AUVs (with low-end processors) to apply NMPC.

5. Verification

In this section, the proposed distributed NMPC implementation is verified by simulating the Falcon AUV to track desired trajectories. The AUV model parameters can be found in Proctor (2014). All the simulations are performed on a personal laptop computer (CPU: Intel(R) Core(TM) i7-3520M: 2.90 GHz 2.90 GHz; RAM: 4.00 GB). The embedded MATLAB function *fmincon* is used as the NLP solver in the simulations.

5.1. Parameter selection

Two trajectories are used to test the proposed distributed implementation (DMPC). The first trajectory represents a sinusoidal path (Case I) in the horizontal plane and is defined as follows:

$$p(t) = \begin{cases} x_d = 0.5t \\ y_d = \sin(0.5t) \end{cases} \tag{27}$$

The second test trajectory is an eight-shaped trajectory (Case II) defined as follows,

$$p(t) = \begin{cases} x_d = \sin(-0.5t) \\ y_d = \sin(0.25t) \end{cases} \tag{28}$$

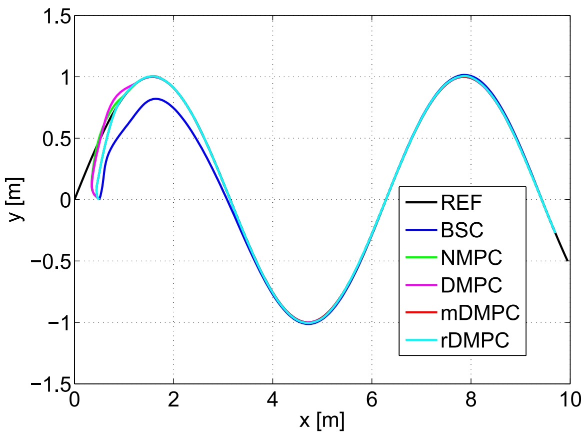


Fig. 1. The AUV successfully tracks the desired trajectory with different control algorithms — Case I. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

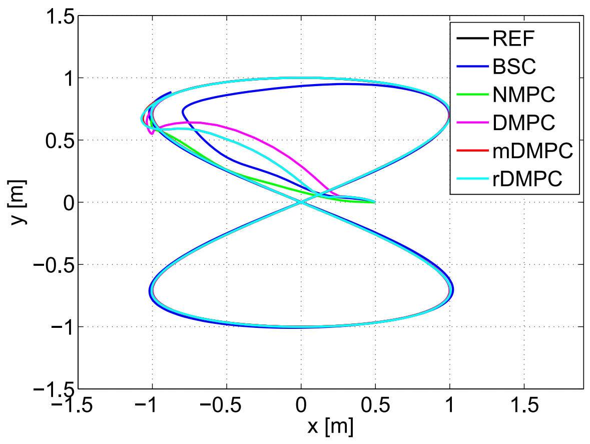


Fig. 2. The AUV successfully tracks the desired trajectory with different control algorithms — Case II. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

For the NMPC tracking controller we use the following parameters: the sampling period $\delta = 0.1$ [s]; the prediction horizon $T = 5\delta$; the weighting matrices $Q = \text{diag}(10^5, 10^5, 10^3, 10^2, 10^2, 10^2)$, $R = \text{diag}(10^{-4}, 10^{-4}, 10^{-4}, 10^{-4})$ and $P = \text{diag}(10^3, 10^3, 10^2, 10, 10, 10)$; the limit on each thruster is 500 [N]. The control gains $\mathbf{K}_p = \mathbf{K}_d = \text{diag}(1, 1, 1)$; and the initial condition $\mathbf{x}(0) = [0.5, 0, 0, 0, 0, 0]^{\text{T}}$. For the proposed real-time distributed implementation, we set time constraint $t_{\max} = 0.1$ [s], initial maximum iteration number $i_{\text{SQP,max}} = 500$, $\alpha = 0.5$ and $\Delta i_1 = \Delta i_2 = 10$.

5.2. Tracking performance

The simulation results are plotted in Figs. 1–2. In each plot, the blue curve is the simulated AUV trajectory using the auxiliary backstepping control (BSC); the green curve is the AUV trajectory with the centralized NMPC implementation; the magenta curve is the vehicle trajectory with the parallel distributed implementation (DMPC); the red curve is with the modified sequential distributed implementation (mDMPC); the cyan curve is the AUV trajectory generated by the real-time distributed implementation algorithm (rDMPC), and the black curve is the desired trajectory (REF). Generally speaking, similar observations can be made for the two test cases: all the tracking controllers successfully drive the vehicle converging to the desired trajectories, which demonstrates the control stability for each closed-loop system controlled by different methods. However, as mentioned several times, this important stability property is not automatically ensured by the optimality of the solution. The stability obtained for NMPC and DMPC owe to the good selection of parameters (i.e., the prediction horizon and the weighting matrices).

Figs. 3 and 4 exemplify the lost stability. If the parameters are badly selected, the AUV controlled by NMPC and DMPC may not converge to the desired trajectory. As can be seen from these two figures, when we choose a bad prediction horizon, the AUV trajectories controlled by NMPC (for $N = 1$) and DMPC (for both $N = 1$ and $N = 2$) will diverge. Similar instability can occur if the weighting matrices are badly chosen. In contrast, the mDMPC and rDMPC demonstrate the guaranteed closed-loop stability in both cases, which highlights their main advantages. Nevertheless, for tracking fixed types of desired trajectories, we are usually able to find a set of well-tuned parameters. In AUV applications that involve tracking of arbitrary trajectories (such as fully autonomous driving), it is better to have the theoretical guarantees.

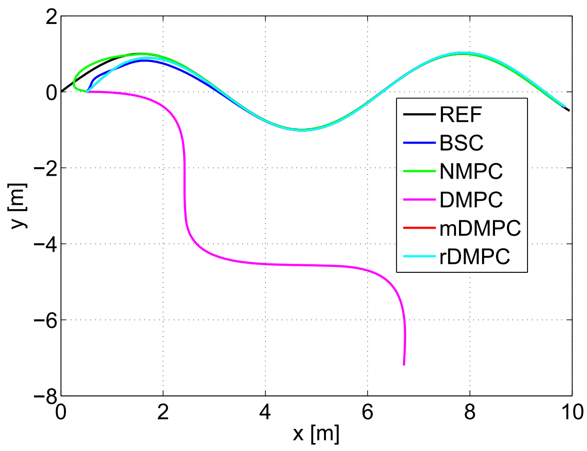


Fig. 3. The lost stability due to badly chosen parameters — Case I with N = 2.

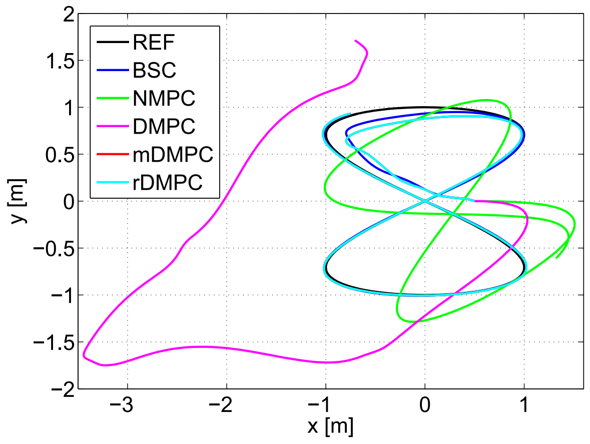


Fig. 4. The lost stability due to badly chosen parameters — Case II with N = 1.

Further comparing the tracking performance (for stable cases), we find that with the MPC controllers the AUV converges faster than using the auxiliary BSC controller. This is because the MPC leverages online optimization to search for the best possible solution, while the control gains $\mathbf{K}_p$ and $\mathbf{K}_d$ for BSC are selected to be small for a large guaranteed region of attraction. As discussed in Remark 1, since we introduce certain conservativeness into

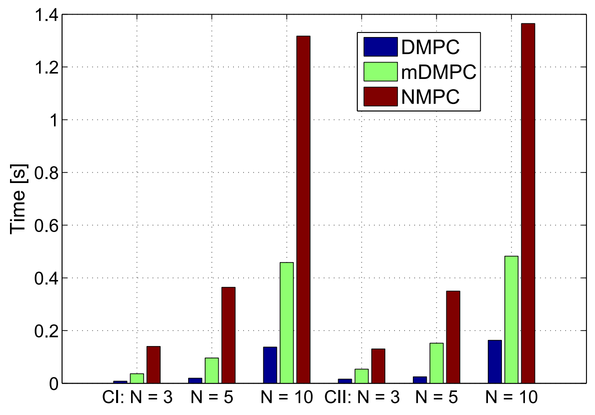


Fig. 5. Average computation time per update.

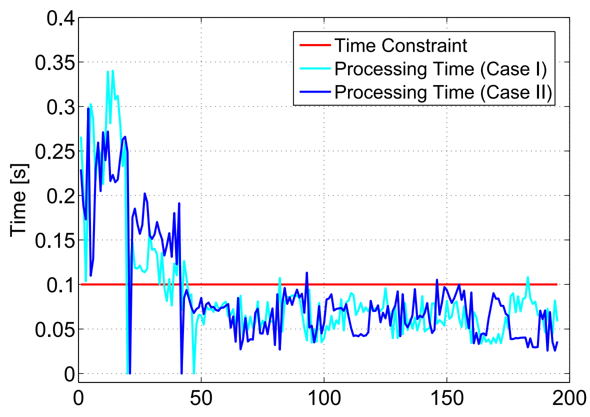


Fig. 6. Processing time for the proposed real-time distributed implementation.

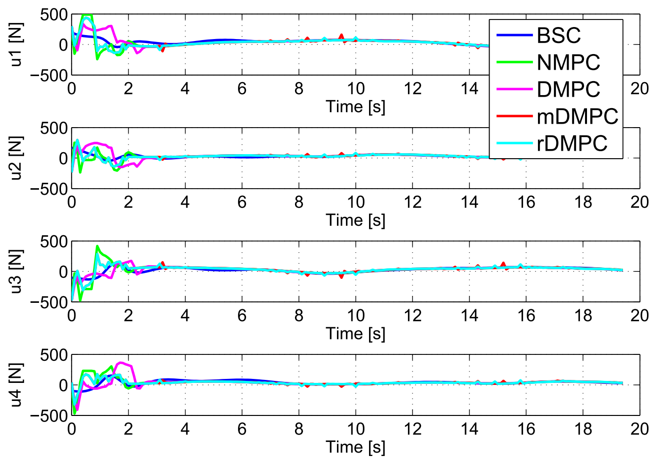


Fig. 7. The control command for each thruster — Case I.

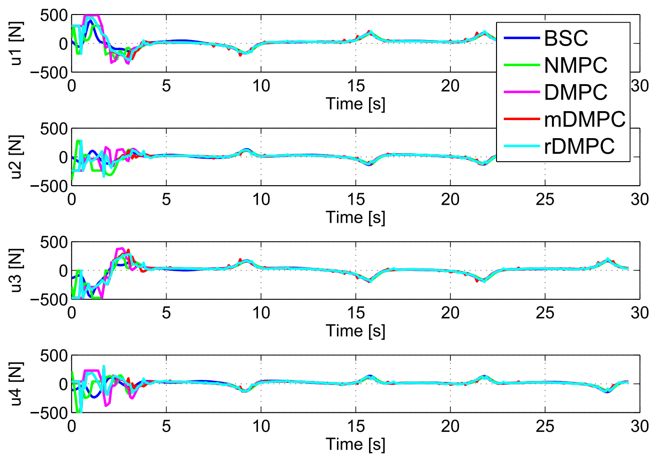


Fig. 8. The control command for each thruster — Case II.

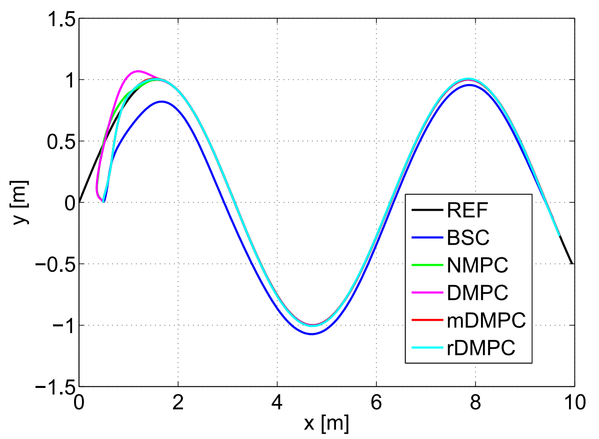


Fig. 9. The MPC controllers are able to steer the AUV to track the desired trajectory in presence of large model error and external disturbances — Case I.

the distributed implementation, the centralized NMPC should outperform the DMPC, which can be observed from Figs. 1 and 2. However, if we compare the average computing time for each implementation, as shown in Fig. 5, the distributed implementations demonstrate significant improvement in terms of algorithmic efficiency. The processing time for the real-time distributed implementation (rDMPC) is plotted in Fig. 6, which shows that the time constraint can be respected during the control.

The control commands for the thrusters are recorded in Figs. 7 and 8. We observe that in the beginning of the tracking, the MPC controllers make more aggressive moves than the BSC controller to get the fastest possible convergence. The BSC with a fixed control gain cannot fully use the onboard thrust capability, which reveals the advantage of MPC. As expected, all the control commands stay within the permitted range of magnitude.

5.3. *Robustness test*

In real world AUV application scenarios, disturbances and uncertainties are ubiquitous. To test the robustness, we set 20% parametric error in the obtained dynamic model of Falcon (based on which the controllers calculate the control commands) and then check the tracking performance in the presence of cyclic disturbances: $\mathbf{w} = [20\sin(t), 30\cos(0.5t), 20\sin(0.2t)]^{\mathrm{T}}$.

From the simulation results illustrated in Figs. 9–10, we observe that all MPC controllers lead the AUV converging to the desired trajectories in the presence of parametric uncertainties and external disturbances, while the BSC exhibits visible tracking error. This demonstrates the prominent robustness against parametric error and external disturbances.

For the DMPC implementation, the computation of control command for each subsystem can be performed in parallel. For each subproblem we need to have an assumed state trajectory $\hat{\mathbf{x}}$ constructed using the solutions of neighboring subsystems (16). Ideally, each subsystem uses the same assumed trajectory. However, due to communication latency, the solutions of neighboring

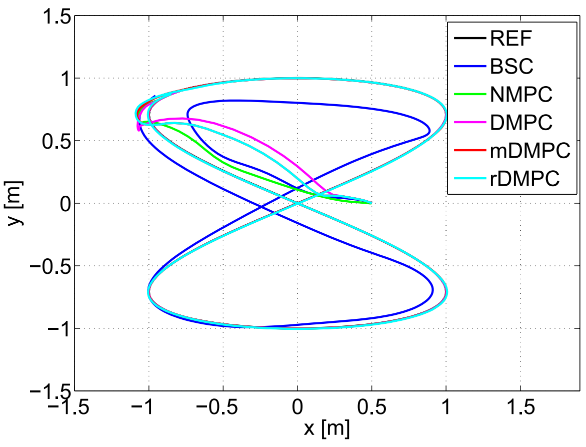


Fig. 10. The MPC controllers are able to steer the AUV to track the desired trajectory in presence of large model error and external disturbances — Case II.

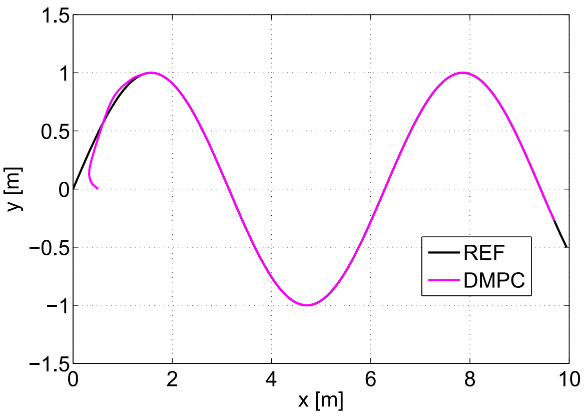


Fig. 11. The AUV successfully tracks the desired trajectory by DMPC with mismatched assumed trajectories — Case I.

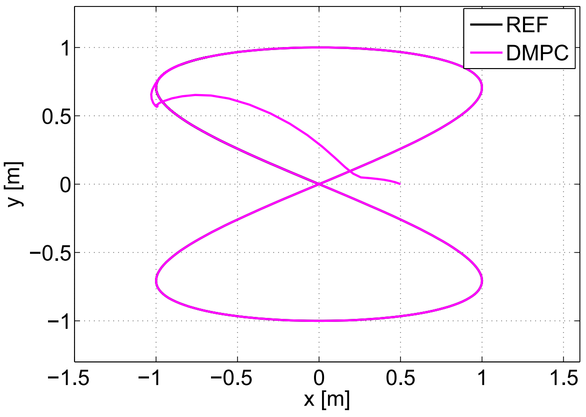


Fig. 12. The AUV successfully tracks the desired trajectory by DMPC with mismatched assumed trajectories — Case II.

subsystems may not be updated in time. This will result in a mismatch in the assumed trajectories, which may influence the control performance. We also test the robustness of the DMPC against such mismatch. In Figs. 11–12, we simulate the tracking control in the situation that each subsystem constructs the assumed trajectory using the solution of its neighbors for the previous time. The simulation results demonstrate satisfactory tracking performances, which implies excellent robustness.

6. Conclusions

In this paper, we have investigated the nonlinear model predictive control for the trajectory tracking application of an autonomous underwater vehicle. The dynamic properties of the vehicle motion were particularly exploited and a novel distributed implementation strategy was proposed to alleviate the computational burden. Extensive simulation studies revealed the effectiveness and robustness of the proposed method and highlighted the advantage of the NMPC tracking control.

References

Buskens, C., & Maurer, H. (2000). SQP-methods for solving optimal control problems with control and state constraints: adjoint variables, sensitivity analysis and real-time control. *Journal of Computational and Applied Mathematics*, *120*, 85–108.

Do, K., Jiang, Z., Pan, J., & Nijmeijer, H. (2004). A global output-feedback controller for stabilization and tracking of underactuated ODIN: A spherical underwater vehicle. *Automatica*, *40*, 117–124.

Fossen, T. (2002). *Marine control systems: Guidance, navigation and control of ships, rigs and underwater vehicles*. Trondheim, Norway: Marine Cybernetics.

Guerreino, B., Silvestre, C., Cunha, R., & Pascoal, A. (2014). Trajectory tracking nonlinear model predictive control for autonomous surface craft. *IEEE Transactions on Control Systems Technology*, *22*, 2160–2175.

Khalil, H. (1996). *Nonlinear systems*. New York: Prentice Hall.

Li, H., & Shi, Y. (2014). Event-triggered robust model predictive control of continous-time nonlinear systems. *Automatica*, *50*, 1507–1513.

Li, H., & Yan, W. (2017). Model predictive stabilization of constrained underactuated autonomous underwater vehicles with guaranteed feasibility and stability. *IEEE/ASME Transactions on Mechatronics*, *22*, 1185–1194.

Lindegaard, K. (2003). *Acceleration feedback in dynamic positioning systems* (Ph.D. thesis), Norwegian University of Science and Technology.

Mahmoud, M. (2018). *Advanced control design with application to electromechanical systems*. Brazil: Elsevier.

Mayne, D. (2014). Model predictive control: recent developments and future promise. *Automatica*, *50*, 2967–2986.

McGovern, L. (2000). *Computational analysis of real-time convex optimization for control system* (Ph.D. thesis), Cambridge, MA, USA: Massachusetts Institute of Technology.

Naeem, W., Sutton, R., & Ahmad, S. (2003). Pure pursuit guidance and model predictive control of an autonomous underwater vehicle for cable tracking. In *World maritime technology conference*, San Francisco, Calnifornia, USA.

Proctor, A. (2014). *Semi-automous guidance and control of a Saab SeaEye Falcon ROV* (Ph.D. thesis), University of Victoria.

Repoulias, F., & Papadopoulos, E. (2007). Plannar trajectory planning and tracking control design for underactuated AUVs. *Ocean Engineering*, *34*, 1650–1667.

Richter, S., Jones, C., & Morari, M. (2012). Computational complexity certification for real-time MPC with input constraints based on the fast gradient method. *IEEE Transactions on Automatic Control*, *57*, 1391–1403.

Shen, C., Shi, Y., & Buckham, B. (2016). Nonlinear model predictive control for trajectory tracking of an AUV: A distributed implementation. In *Proceedings of the IEEE 55th conference on decision and control*, Las Vegas, USA.

Shen, C., Shi, Y., & Buckham, B. (2017a). Integrated path planning and tracking control of an AUV: A unified receding horizon optimization approach. *IEEE/ASME Transactions on Mechatronics*, *22*, 1163–1173.

Shen, C., Shi, Y., & Buckham, B. (2017b). Lyapunov-Based model predictive control for dynamic positioning of autonomous underwater vehicles. In *Proceedings of the IEEE international conference on unmanned systems*, Beijing, China.

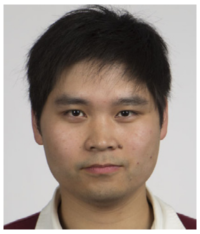
Shen, C., Shi, Y., & Buckham, B. (2017c). Modified C/GMRES algorithm for fast nonlinear model predictive tracking control of AUVs. *IEEE Transactions on Control Systems Technology*, *25*, 1896–1904.

Shen, C., Shi, Y., & Buckham, B. (2018). Trajectory tracking control of an autonomous underwater vehicle using Lyapunov-based model predictive control. *IEEE Transactions on Industrial Electronics*, *65*.

Shi, Y., Shen, C., Fang, H., & Li, H. (2017). Advanced control in marine mechatronic systems: A survey. *IEEE/ASME Transactions on Mechatronics*, *22*, 1121–1131.

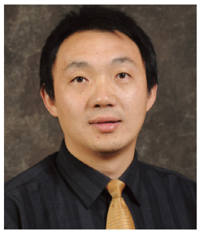
Soylu, S., Buckham, B., & Podhorodeski, R. (2008). A chattering-free sliding-mode controller for underwater vehicles with fault-tolerant infinity-norm thrust allocation. *Ocean Engineering*, *35*, 1647–1659.

Soylu, S., Proctor, A., Podhorodeski, R., Bradley, C., & Buckham, B. (2015). Precise trajectory control for an inspection class ROV. *Ocean Engineering*, *111*, 508–523.



Chao Shen received the B.E. degree in automation engineering and M.Sc. in control science and engineering from Northwestern Polytechnical University, Xi'an, China in 2009 and 2012, respectively, and the Ph.D. degree in mechanical engineering from the University of Victoria, Victoria, Canada, in 2018.

He is currently working as a Research Scientist in the Robotics Division, AltumView Systems Inc., Port Moody, Canada. His main research interests include model predictive control, robotics, mechatronics, deep learning and computer vision.



Yang Shi received the Ph.D. degree in electrical and computer engineering from the University of Alberta, Edmonton, AB, Canada, in 2005. From 2005 to 2009, he was a Faculty Member with the Department of Mechanical Engineering, University of Saskatchewan, Saskatoon, SK, Canada. He is currently a Professor with the Department of Mechanical Engineering, University of Victoria, Victoria, BC, Canada. He was a Visiting Professor with the University of Tokyo, Tokyo, Japan, in 2013. His current research interests include networked and distributed systems, model predictive control, industrial cyber–physical systems, mechatronics, and energy systems.

Dr. Shi was a recipient of the University of Saskatchewan Student Union Teaching Excellence Award in 2007, the Faculty of Engineering Teaching Excellence Award at the University of Victoria in 2012, the JSPS Invitation Fellowship (short-term), the 2015 Craigdarroch Silver Medal for Excellence in Research at the University of Victoria, and the Humboldt Research Fellowship (for experienced researchers) in 2017. He is currently a Co-Editor-in-Chief of the IEEE Transactions on Industrial Electronics. He also serves as an Associate Editor for Automatica, the IEEE Transactions on Control Systems Technology, the IEEE/ASME Transactions on Mechatronics, the IEEE Transactions on Cybernetics, and the ASME Journal of Dynamic Systems, Measurement, and Control. He is a Fellow of ASME and CSME, and a P.Eng. in British Columbia, Canada.