Learning-based Nonlinear Model Predictive Control of Reconfigurable Autonomous Robotic Boats: Roboats

Learning-based Nonlinear Model Predictive Control of Reconfigurable Autonomous Robotic Boats: Roboats

Erkan Kayacan [1], Shinkyu Park[2,3], Carlo Ratti [2] and Daniela Rus [3]

***Abstract*— This paper presents a Learning-based Nonlinear Model Predictive Control (LB-NMPC) algorithm for reconfigurable autonomous vessels to facilitate high-accurate path tracking. Each vessel is designed to latch to a pre-defined point of another vessel that allows the vessels to form a rigid body. The number of possible configurations of such vessels exponentially grows as the total number of vessels increases, which imposes a technical challenge in modeling and identification. In this work, we propose a framework consisting of a real-time parameter estimator and a feedback control strategy, which is capable of ensuring high-accurate path tracking for any feasible configuration of vessels. Novelty of our method is in that the parameter is estimated on-line and adjusts control parameters (e.g., cost function and dynamic model) simultaneously to improve path-tracking performance. Through experiments on different configurations of connected-vessels, we demonstrate stability of our proposed approach and its effectiveness in high-accuracy in path tracking.**

I. Introduction

Self-driving car technology encourages researchers for having a fleet of autonomous vessels to change our cities and their waterways. These reconfigurable autonomous vessels can be used for many purposes, such as food delivery, infrastructure and garbage collection. They can (i) form floating food markets and become pop-ups stalls that appear on waterways' edges to provide crates of fresh produce, and (ii) form temporary floating structures like bridges, concert stages, and public squares for events on waterways (See Fig. 1(a) for an illustration). Thus, they activate the canals while tapping into the resources located on the pervasive regional network of waterways. Moreover, large trash trucks on streets cause many problems, such as pollution and noise. Reconfigurable autonomous vessels can serve as floating dumpsters that can collect garbage and transfer waste, where residents deposit trash on curbs for collection (See Fig. 1(b) for an illustration). To achieve all these tasks, a fleet of reconfigurable autonomous vessels require modular boat platforms that can latch together to create variable size and form [1]–[4].

Traditionally developing control algorithms for autonomous vessels is challenging in that the controller should maintain high-precision path-tracking performance in the presence of matched and mismatched disturbances, such as current and waves [5]. In addition, unmodeled parameters in dynamic models introduce additional difficulties in vessel control, and this requires control methods to adaptively change control strategies based on sensor measurements of vessels [7]–[9]. Furthermore, modular boat platforms, which can latch together, create a new control problem. The challenges in controlling reconfigurable vessels stem from dynamically changing shape and number of connected-vessels, which would dramatically change dynamic models of the vessels and cause instability in path tracking [10], [11]. This motivates us to develop and apply a Learning-based Nonlinear Predictive Control (LB-NMPC) algorithm for reconfigurable autonomous vessels.

(a) (b)

Fig. 1. Concept of autonomous connected-vessels (Roboats): Individual units can (a) tessellate together to form floating stages and public squares on waterways and (b) serve as floating garbage bins that autonomously transfer waste.

In this paper, we describe the design and implementation of reconfigurable autonomous vessels with experiment results demonstrating their ability to track a given path in arbitrary configurations. The key innovations of our work include: (i) creation of the first hardware prototype for connected-vessels, (ii) development of a generalized method for controlling arbitrary configurations of connected-vessels, and (iii) a real-time iteration for LB-NMPC algorithm to reduce the required computational burden.

The problem is formulated in Section II. The novel LB-NMPC algorithm for reconfigurable autonomous vessels and proposed real-time iteration scheme are given in Section III. Roboat description and experimental results are presented in Section IV. Section V concludes the paper.

II. Problem Formulation

In this paper, we consider a problem of designing a learning-based feedback control algorithm for connected autonomous vessels. The vessels are designed to connect to one another to form a rigid body as illustrated in Fig. 1. To describe the problem, we explain a dynamic model and parameter identification/feedback control of the connected-vessels.

*This work was supported by the grant from the Amsterdam Institute for Advanced Metropolitan Solutions (AMS) in the Netherlands.

[1] Erkan Kayacan is with the School of Mechanical & Mining Engineering University of Queensland, Brisbane, Australia. Email:e.kayacan@uq.edu.au

[2] Shinkyu Park and Carlo Ratti are with the Senseable City Laboratory (SCL), Massachusetts Institute of Technology, Cambridge, MA, USA. Emails:{shinkyu, ratti}@mit.edu

[3] Shinkyu Park and Daniela Rus are with the Computer Science & Artificial Intelligence Laboratory (CSAIL), Massachusetts Institute of Technology, Cambridge, MA, USA. Emails:{shinkyu, rus}@mit.edu

A. Modeling of Connected-Vessels

The moving coordinate frame fixed at the connected-vessels is called the body-fixed reference frame. The origin of the body-fixed frame is chosen to coincide with the center of the gravity. Moreover, the motion of the body-fixed frame is described relative to an inertial reference frame. The position and orientation of the connected-vessels described relative to the inertial reference frame while the linear and angular velocities of the connected-vessels can be expressed in the body-fixed coordinate system (see Fig. 2 for an illustration). Finally, the kinematic equation is written as follows:

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

where $\boldsymbol{\eta} = [E, N, \psi]^T \in \mathbb{R}^3$ is the vector for position and heading angle of the connected-vessels in the inertial frame, $\boldsymbol{\nu} = [u, v, r]^T \in \mathbb{R}^3$ is the vector for the velocities on the body-fixed frame, and $\boldsymbol{R}(\psi) = [\cos\psi, -\sin\psi, 0; \sin\psi, \cos\psi, 0; 0, 0, 1]^T$ is the transformation matrix.

Inspired by the approach described in [10], we adopt a simple approximate model that varies only on the number of connected vessels α. The nonlinear connected-vessels dynamics are described by the following differential equations [12]:

$$\mathbf{M}\dot{\boldsymbol{\nu}} + \mathbf{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{D}\boldsymbol{\nu} = \mathbf{A}\boldsymbol{\tau} \tag{2}$$

where $\mathbf{M} \in \mathbb{R}^{3\times3}$ is the positive-definite symmetric inertia matrix, $\boldsymbol{\nu} = [u, v, r]^T \in \mathbb{R}^3$ are the velocities on the body-fixed frame, $\mathbf{C}(\boldsymbol{\nu}) \in \mathbb{R}^{3\times3}$ is the skew-symmetric Coriolis matrix, $\mathbf{D} \in \mathbb{R}^{3\times3}$ is the positive-semi definite drag matrix, $\mathbf{A} \in \mathbb{R}^{3\times3}$ is the positive definite diagonal matrix and $\boldsymbol{\tau} = [\tau_u, \tau_v, \tau_r]^T \in \mathbb{R}^3$ is the forces and torque applied to the connected-vessels.

The inertia matrix is written as follows:

$$\mathbf{M} = \mathrm{diag}\,(m_{11}, m_{22}, m_{33}) \tag{3}$$

Considering the fact that the origin and the center of mass of the connected-vessels are the same point [13], the Coriolis matrix is written as follows:

$$\mathbf{C}(\boldsymbol{\nu}) = \begin{bmatrix} 0 & 0 & -m_{22}v \\ 0 & 0 & m_{11}u \\ m_{22}v & -m_{11}u & 0 \end{bmatrix} \tag{4}$$

The connected-vessels always moves at low speed (e.g., maximum speed is 6 $\mathrm{km\,h^{-1}}$) so that the drag matrix can be described by a linear damping term. Moreover, the vessel platform is symmetric with respect to longitudinal and lateral axes in the body-fixed frame. Therefore, the drag matrix is written as follows:

$$\mathbf{D} = \mathrm{diag}\,(\zeta_u, \zeta_v, \zeta_r) \tag{5}$$

where $\boldsymbol{\zeta} = [\zeta_u, \zeta_r, \zeta_r,]^T \in \mathbb{R}^3$ is the damping coefficients vector.

The matrix $\boldsymbol{A}$ is written as follows:

$$\boldsymbol{A} = \mathrm{diag}\left(\frac{1}{\alpha}, \frac{1}{\alpha}, \frac{1}{\alpha^2}\right) \tag{6}$$

where α is a parameter for the number of connected-vessels. From the forces $\{f_i\}_{i=1}^{4n} \in \mathbb{R}^{4n}$ applied to the propellers

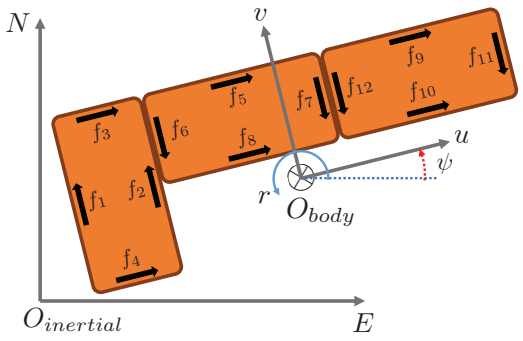


Fig. 2. An example of connected system consisting of 3 vessels.

on the connected-vessels, we define the forces and torque $\boldsymbol{\tau} = (\tau_u, \tau_v, \tau_r) \in \mathbb{R}^3$ applied at the center of mass (CM) as follows:

$$\begin{pmatrix} \tau_u \\ \tau_v \end{pmatrix} = \sum_{i=1}^{4n} f_i \quad \text{and} \quad \tau_r = \sum_{i=1}^{4n} r_i \times f_i \tag{7}$$

where r_i is the vector from CM to the point at which f_i is applied.

As it is clear from (2), the parameters $\boldsymbol{m}$ and $\boldsymbol{\zeta}$ defined by $\boldsymbol{m} = (m_{11}, m_{22}, m_{33})$ and $\boldsymbol{\zeta} = (\zeta_u, \zeta_v, \zeta_r)$ can be found through analysis of experiment data obtained using a single vessel ($\alpha = 1$). For notational convenience, given parameters $(\boldsymbol{m}, \boldsymbol{\zeta}, \alpha)$, we refer to the Connected-Vessels Dynamic Model (2) as $CVDM(\boldsymbol{m}, \boldsymbol{\zeta}, \alpha)$. The variables are represented in Table I.

Remark 1 (Discovery of Platform Configuration): To define the transformation (7), we need the information on how the vessels are connected in the platform from which we can infer the orientations of the force vectors f_i and the values of the parameters r_i. This problem is known to be configuration discovery [14], [15] in modular robotics. Due to space limit, our work in this paper focuses on designing learning-based control scheme, and we leave integration of algorithms and hardware for configuration discovery as a future plan.

We define the state of the platform by the pose (E, N, ψ) of CM in the inertial frame in conjunction with the velocity (u, v, r) in the body-fixed frame defined at CM (see Fig. 2 for an illustration). The system model describes how the full state $\boldsymbol{x} = (\boldsymbol{\eta}, \boldsymbol{\nu}) = (E, N, \psi, u, v, r) \in \mathbb{R}^6$ changes in response to applied control input $\boldsymbol{\tau} = (\tau_u, \tau_v, \tau_r) \in \mathbb{R}^3$.

In the rest of the paper, we denote a nonlinear system model as

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

where $\boldsymbol{x} \in \mathbb{R}^{n_x}$ is the state vector, $\boldsymbol{\tau} \in \mathbb{R}^{n_\tau}$ is the control input, $\alpha \in \mathbb{R}$ is a parameter, $f(\cdot,\cdot,\cdot) : \mathbb{R}^{n_x+n_\tau+1} \longrightarrow \mathbb{R}^{n_x}$ is the continuously differentiable state update function and $f(0, 0, \alpha) = 0\ \forall t$. The derivative of x with respect to t is denoted by $\dot{x} \in \mathbb{R}^{n_x}$.

Similarly, a nonlinear measurement model denoted $\boldsymbol{y}(t)$ can be described with the following equation:

$$\boldsymbol{y}(t) = h(\boldsymbol{x}(t), \boldsymbol{\tau}(t), \alpha) \tag{9}$$

where $h : \mathbb{R}^{n_x+n_\tau+1} \longrightarrow \mathbb{R}^{n_y}$ is the measurement function, which describes the relation between the variables of the

system model and the measured outputs of the real-time system.

The state, input and output vectors are respectively denoted as follows:

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{\eta} & \boldsymbol{\nu} \end{bmatrix}^T = \begin{bmatrix} E & N & \psi & u & v & r \end{bmatrix}^T \tag{10a}$$

$$\boldsymbol{\tau} = \begin{bmatrix} \tau_u & \tau_v & \tau_r \end{bmatrix}^T \tag{10b}$$

$$\boldsymbol{y} = \begin{bmatrix} E & N & \psi & r & \tau_u & \tau_v & \tau_r \end{bmatrix}^T \tag{10c}$$

B. Problem Description

As we mentioned in Section I, one key technical challenge in reconfigurable vessels control is that computing precise value of α for every possible vessels configuration is impractical, and high-precision path-tracking performance is hard to achieve without accurate estimates of the parameters. To address this challenge, we consider a problem of designing a model parameter estimator and an optimal feedback controller.

We begin with the following description of a parameter estimation problem: Given output $\boldsymbol{y}(t) = h(\boldsymbol{x}(t), \boldsymbol{\tau}(t))$ of the platform and the control inputs to the propellers $\{f_i(t)\}_{i=1}^{4n}$ over the time horizon $[0,T)$, the parameter identification problem is to find an estimator $\mathscr{E}$ that minimizes

$$\mathscr{J}^{\mathscr{E}}\left(\{\boldsymbol{y}(t), \{f_i(t)\}_{i=1}^{4n} \,|\, t \in [0,T)\}\right) = \int_0^T \|\boldsymbol{y}(t) - h\left(\hat{\boldsymbol{x}}(t)\right)\|^2 \,\mathrm{d}t \tag{11}$$

subject to the following constraints:

dynamic model $CVDM(\boldsymbol{m}, \boldsymbol{\zeta}, \hat{\alpha})$

$$(\hat{\boldsymbol{x}}(t), \boldsymbol{m}, \boldsymbol{\zeta}, \hat{\alpha}) = \mathscr{E}\left(\{\boldsymbol{y}(s), \{f_i(s)\}_{i=1}^{4n} \,|\, s \in [0,t)\}\right), \ \forall t \in [0,T)$$

To maneuver the connected-vessels, we need to assign proper control inputs to the propellers. For this purpose, we consider the following optimal path tracking problem to find a control law $\mathscr{U}$: Given the model parameter $(\boldsymbol{m}, \boldsymbol{\zeta}, \hat{\alpha})$ and sets $\mathbb{X}$, $\mathbb{V}$, $\mathbb{F}$, find $\mathscr{U}$ that minimizes

$$\mathscr{J}^{\mathscr{U}}(\boldsymbol{x}_0) = \int_0^T \left[\left\|\boldsymbol{x}(t) - \boldsymbol{x}_{ref}(t)\right\|^2 + \sum_{i=1}^{4n} |f_i(t)|^2 \right] \mathrm{d}t \tag{13}$$

subject to the following constraints:

$$\text{dynamic model } CVDM(\boldsymbol{m}, \boldsymbol{\zeta}, \hat{\alpha}), \ \boldsymbol{x}(0) = \boldsymbol{x}_0 \tag{14a}$$

$$(E, N, \psi) \in \mathbb{X} \tag{14b}$$

$$(u, v, r) \in \mathbb{V} \tag{14c}$$

$$f_i = \mathscr{U}_i(\boldsymbol{x}(t)) \in \mathbb{F}, \ \forall i \in \{1, \cdots, 4n\} \tag{14d}$$

The sets $\mathbb{X}$ and $\mathbb{V}$ can be regarded as safety constraint sets, which limit respectively the movement and speed of the connected-vessels, and the set $\mathbb{F}$ limits the force that each propeller generates.

Note that in the implementation of the LB-NMPC algorithm described in Section III, the parameter used to find a solution to (13) comes from the parameter estimator $\mathscr{E}$ and control inputs generated by the control law $\mathscr{U}$ are fed into the parameter estimator to refine the parameter estimates. For this reason, to address the challenges in the control of connected-vessels, we need to find a joint solution to both the parameter estimation and control problems. We summarize the main problem addressed in this paper as follows:

Problem 1: Find a parameter estimator $\mathscr{E}$ and a feedback controller $\mathscr{U}$ that, respectively, minimizes (11) and (13), where at each time $t \in [0,T)$, the parameter α for the model $CVDM(\boldsymbol{m}, \boldsymbol{\zeta}, \alpha)$ is determined by $\mathscr{E}$ given output of the platform $\{\boldsymbol{y}(s) \,|\, s \in [0,t)\}$.

TABLE I
NOMENCLATURE

$\boldsymbol{R}(\cdot)$,	Rotation matrix.
$\boldsymbol{M}, \boldsymbol{C}(\cdot), \boldsymbol{D}$	Inertia, Coriolis and Drag matrices.
$\boldsymbol{\eta}, \boldsymbol{\nu}$	Pose and velocity vectors.
E, N, ψ	East, North, heading angle.
u, v, r	Longitudinal, lateral and angular velocities.
$\boldsymbol{\tau}$	Control vector.
τ_u, τ_v, τ_r	Forces and torque.
$\boldsymbol{B}$	Actuator configuration matrix.
$f_i, 4n$	Forces to thusters and number of thrusters.
$\boldsymbol{x}, \boldsymbol{x}_r$	State and reference state vectors.
$\alpha, \boldsymbol{A}$	Parameter and matrix for parameter.
$\hat{\boldsymbol{x}}, \hat{\alpha}$	Estimated state vector and estimated parameter.
$\boldsymbol{y}, \boldsymbol{y}_m$	System output and measurements vectors.
$f(\cdot,\cdot,\cdot)$	System model.
$h(\cdot,\cdot)$	Measurement model.
N	Estimation and control horizons.
$\boldsymbol{H}_k, \boldsymbol{H}_N$	Inverses of measurements and process noise covariance matrices for NMHE.
$\boldsymbol{Q}_k, \boldsymbol{Q}_N, \boldsymbol{R}$	Weighting matrices for NMPC.
$\|\cdot\|^2$	Euclidean vector norm.

III. Learning-based NMPC of Reconfigurable Autonomous Vessels

In this paper, we propose an LB-NMPC algorithm, which consists of coordination, moving horizon estimation, and model predictive control to find a solution to Problem 1. The key idea behind our solution is to adopt a coordinated control scheme: Among the group of connected vessels, the scheme assigns one vessel as a coordinator for which the coordinator computes appropriate control inputs to all propellers in the platform. This approach allows us to simplify the feedback control design. The diagram in Fig. 3 depicts the interplay between these three modules in the algorithm, while the real-time LB-NMPC algorithm is summarized in Algorithm 1.

A. Coordination

We proceed with re-formulating the cost functional (13) in terms of the forces and torque applied at CM. Based on (7) and assuming that the connected-vessels form a rigid body, we can represent $\boldsymbol{\tau}$ as a linear function of the forces $\{f_i\}_{i=1}^{4n}$ that the propellers exert as follows:

$$\boldsymbol{\tau} = \boldsymbol{B}\left(f_1, \ldots, f_{4n}\right)^T \tag{15}$$

where $\boldsymbol{B}$ in a $3 \times 4n$ known matrix that can be derived from (7). Assuming that $\boldsymbol{B}$ has the full row rank, we can find a particular pseudo inverse matrix $\boldsymbol{B}^\dagger$ defined as $\boldsymbol{B}^\dagger = \boldsymbol{B}'(\boldsymbol{B}\boldsymbol{B}')^{-1}$. Note that given $\boldsymbol{\tau}$, the pseudo inverse matrix $\boldsymbol{B}^\dagger$ finds the *minimum-energy* propeller inputs $\{f_i\}_{i=1}^{4n}$: The propeller inputs determined by $[f_1, \cdots, f_{4n}]^T = \boldsymbol{B}^\dagger \boldsymbol{\tau}$ minimize

$$\min_{\{f_i\}_{i=1}^{4n}} \sum_{i=1}^{4n} |f_i|^2 \qquad \text{subject to (15)} \tag{16}$$

Using the transformation matrix $\boldsymbol{B}$ and its pseudo inverse $\boldsymbol{B}^{\dagger}$, we cast the cost function (13) as follows:

$$\mathscr{J}_r^{\mathscr{U}}(\boldsymbol{x}_0) = \int_0^T \left[\left\|\boldsymbol{x}(t) - \boldsymbol{x}_{ref}(t)\right\|^2 + \left\|\boldsymbol{B}^{\dagger}\boldsymbol{\tau}(t)\right\|^2 \right] \mathrm{d}t \quad (17)$$

In the reformulation, notice that the constraints (14) are unchanged except the constraint on the propeller inputs (14d), which now has the following form:

$$\boldsymbol{\tau} \in \mathbb{F}_{\boldsymbol{\tau}} \quad (18)$$

where $\mathbb{F}_{\boldsymbol{\tau}} = \left\{ \boldsymbol{B} \begin{pmatrix} f_1 & \cdots & f_{4n} \end{pmatrix}^T \middle| f_i \in \mathbb{F},\ \forall i \in \{1, \cdots, 4n\} \right\}$.

Compare to the optimal controller synthesis problem described in Section II-B, the cost function (17) has the smaller and fixed number of control variables. Interestingly, it can be shown that any controller $\mathscr{U}^{\star}$ that minimizes (17) can be used to derive an optimal controller for (13). We summarize the argument in the following proposition.

Proposition 1: Suppose that $\mathscr{U}_r^{\star}$ is the optimal controller that minimizes (17). A controller $\mathscr{U}^{\star}$ determined by $\mathscr{U}^{\star}(\boldsymbol{x}) = \boldsymbol{B}^{\dagger}\mathscr{U}_r^{\star}(\boldsymbol{x}),\ \forall \boldsymbol{x} \in \mathbb{X}$ is optimal to (13).

B. Nonlinear Moving Horizon Estimation

In the LB-NMPC approach for reconfigurable vessels, online parameter estimation is required to generate proper control signals to propellers. It is to be noted that the number of vessels cannot be negative so that online parameters estimator must incorporate constraints on the parameter.

Nonlinear Moving Horizon Estimation (NMHE) is an online optimization-based state estimation method that can handle nonlinear systems and satisfy inequality constraints on the estimated states and parameters [16]. Moreover, NMHE method is robust to initial guesses in contrast to Extended Kalman filter, which might fail due to poor initialization [17]. Additionally, NMHE guarantees local stability, while Extended Kalman filter cannot guarantee any general convergence. NMHE considers a fixed number of measurements in a moving time window and the truncated data before the moving time window were presented by the arrival cost [18].

The considered MHE problem is formulated as follows:

$$\min_{\boldsymbol{x}(t),\alpha} \quad \Gamma(\boldsymbol{x}_{k-N}, \alpha) + \Gamma(\boldsymbol{y}_m(t_i), \boldsymbol{y}(t_i)) \quad (19a)$$

$$\text{s. t.} \quad \dot{\boldsymbol{x}}(t) = f\left(\boldsymbol{x}(t), \boldsymbol{\tau}(t), \alpha\right) \qquad \forall t \in [t_{k-N}, t_k] \quad (19b)$$

$$\boldsymbol{y}(t) = h\left(\boldsymbol{x}(t), \boldsymbol{\tau}(t)\right) \qquad \forall t \in [t_{k-N}, t_k] \quad (19c)$$

$$\boldsymbol{x}_{min} \le \boldsymbol{x}(t) \le \boldsymbol{x}_{max} \qquad \forall t \in [t_{k-N}, t_k] \quad (19d)$$

$$\alpha_{min} \le \alpha \le \alpha_{max} \quad (19e)$$

where the cost function (19a) consists of the arrival and quadratic costs, and is defined as follows:

$$\Gamma(\boldsymbol{x}_{k-N}, \alpha) = \left\| \begin{matrix} \hat{\boldsymbol{x}} - \boldsymbol{x}(t_{k-N}) \\ \hat{\alpha} - \alpha \end{matrix} \right\|^2_{\boldsymbol{H}_N} \quad (20a)$$

$$\Gamma(\boldsymbol{y}_m(t_i), \boldsymbol{y}(t_i)) = \sum_{i=k-N}^{k} \|\boldsymbol{y}_m(t_i) - \boldsymbol{y}(t_i)\|^2_{\boldsymbol{H}_k} \quad (20b)$$

The arrival cost (20a) stands for the measurements before the beginning of the estimation horizon ($t = [t_{0,k-N+1}]$) while the quadratic cost (20b) stands for the measurements within the estimation window $t = [t_{t-N+1,k}]$. The measurements are denoted by $\boldsymbol{y}_m$ and the system output is denoted by $\boldsymbol{y}$. The formulation (19) does not take the input variables $\boldsymbol{\tau}$ into account as optimization variables, while they are used to simulate the system. The estimates for states and parameter are denoted by $\hat{\boldsymbol{x}}$ and $\hat{\alpha}$, respectively, while the estimation horizon is denoted by N. All constraints on states and parameter respectively denoted by $\boldsymbol{x}_{min}$, $\boldsymbol{x}_{max}$, α_{min} and α_{max} are summarized in (19d)-(19e), and used for safety to ensure that the physical limitations and/or validity of models are satisfied. The weighting matrices $\boldsymbol{H}_k$ and $\boldsymbol{H}_N$ are the inverses of the measurement and process noise covariance matrices, respectively. They must be chosen adequately to obtain good state and parameter estimates based on knowledge or prediction of the error distributions. Considering the noise characteristics of sensors given in Section IV-A, the weighting matrix $\boldsymbol{H}_k$ used in the experiments in the order of (10c) is as follows:

$$\begin{aligned} \boldsymbol{H}_k &= \mathrm{diag}\left(\sigma_E^2, \sigma_N^2, \sigma_{\psi}^2, \sigma_r^2, \sigma_{\tau_u}^2, \sigma_{\tau_v}^2, \sigma_{\tau_r}^2\right)^{-1} \\ &= \left(10^{-2} \times \mathrm{diag}\left(5^2, 5^2, 7^2, 1^2, 1^2, 1^2, 1^2\right)\right)^{-1} \end{aligned} \quad (21)$$

Additionally, the weighting matrix H_N used in the experiments in the order of (10a) and α, respectively, is as follows:

$$\begin{aligned} \boldsymbol{H}_N &= \mathrm{diag}(E, N, \psi, u, v, r, \alpha)^{-1} \\ &= \mathrm{diag}(10, 10, 6.28, 1, 1, 10, 0.25)^{-1} \end{aligned} \quad (22)$$

The weighting coefficients for immeasurable variables (e.g., u,v and α) are chosen smaller than measurable variables to obtain a smooth estimates of these variables. Moreover, the lower and upper bounds on the parameter α are defined as follows:

$$0.25 \le \alpha \le 10 \quad (23)$$

C. Nonlinear Model Predictive Control

Nonlinear Model Predictive Control (NMPC) is a family of optimal control algorithms that anticipate systems' behavior

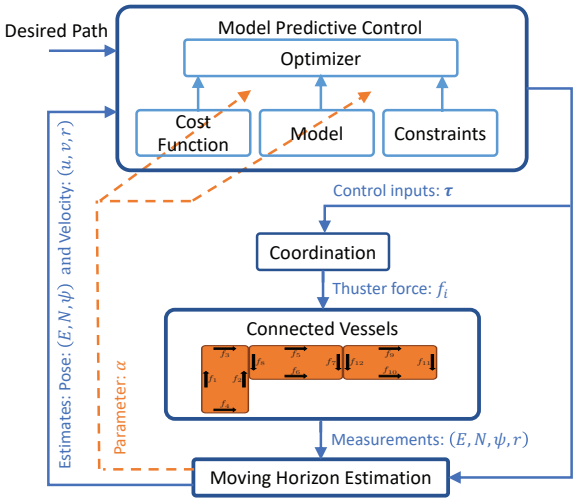


Fig. 3. Schematic diagram for LB-NMPC of reconfigurable autonomous vessels.

in the future by minimizing a given cost function [19], [20]. The main principle is to solve an optimal control problem at every time iteration to obtain a control trajectory $\mathscr{U}$, which guarantees the constraints on system states and inputs [21].

The cost functions consist of least squares error between a given and predicted state trajectories. NMPC solves the following nonlinear constrained optimization problem at every time instant:

$$\min_{x(t),u(t)} \quad \sum_{i=k+1}^{k+N-1} l\Big(\boldsymbol{x}(t_i),\boldsymbol{\tau}(t_i)\Big) + l_N\Big(\boldsymbol{x}_N(t_i),\boldsymbol{\tau}_N(t_i)\Big) \tag{24a}$$

$$\text{s. t.} \quad \boldsymbol{x}(t_k) = \hat{\boldsymbol{x}}(t_k) \tag{24b}$$

$$\alpha = \hat{\alpha}(t_k) \tag{24c}$$

$$\dot{\boldsymbol{x}}(t) = f\big(\boldsymbol{x}(t),\boldsymbol{\tau}(t),\alpha\big) \tag{24d}$$

$$\boldsymbol{x}_{\min} \le \boldsymbol{x}(t) \le \boldsymbol{x}_{\max} \qquad t \in [t_{k+1}, t_{k+N}] \tag{24e}$$

$$\boldsymbol{\tau}_{\min} \le \boldsymbol{\tau}(t) \le \boldsymbol{\tau}_{\max} \qquad t \in [t_{k+1}, t_{k+N-1}] \tag{24f}$$

where the stage and terminal cost functions are defined as follows:

$$l\Big(\boldsymbol{x}(t_i),\boldsymbol{\tau}(t_i)\Big) = \|\boldsymbol{x}_r(t_i) - \boldsymbol{x}(t_i)\|^2_{\boldsymbol{Q}} + \|\boldsymbol{\tau}(t_i)\|^2_{\boldsymbol{R}} \tag{25a}$$

$$l_N\Big(\boldsymbol{x}_N(t_i),\boldsymbol{\tau}_N(t_i)\Big) = \|\boldsymbol{x}_r(t_{k+N}) - \boldsymbol{x}(t_{k+N})\|^2_{\boldsymbol{Q}_N} \tag{25b}$$

where $\boldsymbol{Q} \in \mathbb{R}^{n_x \times n_x}$, $\boldsymbol{R} \in \mathbb{R}^{n_\tau \times n_\tau}$ and $\boldsymbol{Q}_N \in \mathbb{R}^{n_x \times n_x}$ are symmetric positive semi-definite weighting matrices, $\boldsymbol{x}_r$ is the reference state trajectory, $\boldsymbol{x}$ and $\boldsymbol{\tau}$ are respectively the states and inputs, t_k is the current time, N is the prediction horizon, $\hat{\boldsymbol{x}}(t_k)$ is the current estimates for states, $\hat{\alpha}(t_k)$ is the current estimate for the parameter, $\boldsymbol{x}_{\min}$ and $\boldsymbol{x}_{\max}$ are the constraints on the states $\boldsymbol{x}$, whereas $\boldsymbol{\tau}_{\min}$ and $\boldsymbol{\tau}_{\max}$ are the constraints on the control inputs $\boldsymbol{\tau}$. The matrix $\boldsymbol{Q}$ weights the difference between the given and actual state trajectory throughout the prediction horizon while the matrix $\boldsymbol{Q}_N$ weights the difference between the given and actual state trajectory at the end of the prediction horizon.

The weighting matrices $\boldsymbol{Q}_k$ and $\boldsymbol{Q}_N$ in the order of (10a) and $\boldsymbol{R}$ in the order of (10b) for NMPC used in the experiments are respectively chosen as

$$\boldsymbol{Q} = \text{diag}\,(1,1,0.3,0,0,0), \quad \text{and} \quad \boldsymbol{Q}_N = 10 \times \boldsymbol{Q} \tag{26a}$$

$$\boldsymbol{R} = \text{diag}\,(0.5,0.5,2) \times \boldsymbol{A}^2 \tag{26b}$$

It is to be noted that the weighting matrix $\boldsymbol{R}$ changes by time with respect to the estimated parameter α in real-time. The reason is that we want to maintain the same aggressiveness for the control algorithm and thus obtain the similar path tracking performance for any feasible configuration of connected-vessels. Moreover, we take the square of the matrix $\boldsymbol{A}$ in the formulation of the weighting matrix $\boldsymbol{R}$ (26b) due to the least-square formulation in (25a). The constraints on the control inputs $\boldsymbol{\tau}$ used in the experiments are as follows:

$$-40\text{N} \le \tau_u \text{ and } \tau_v \le 40\text{N} \tag{27a}$$

$$-40\text{Nm} \le \tau_r \le 40\text{Nm} \tag{27b}$$

Algorithm 1 LB-NMPC algorithm

Initialize:

$\boldsymbol{x}(t_k) = \boldsymbol{x}(0)$, $\hat{\boldsymbol{x}}(t_k) = \hat{\boldsymbol{x}}(0)$ and $\boldsymbol{\tau}(t_k) = \boldsymbol{\tau}(0)$

Inputs:

$\boldsymbol{y}_m(t_k) = x(t_k), y(t_k), \psi(t_k), r(t_k), \tau_u(t_{k-1}), \tau_v(t_{k-1}), \tau_r(t_{k-1})$

Outputs:

$f_i(t_k)$, $i = 1,\ldots,4n$.

while $k \ge N+1$ **do**

MHE receives the current measurements $\boldsymbol{y}_m(t_k)$.

Solve the MHE (19)

Obtain the state estimate trajectory $\mathscr{X} = [\hat{\boldsymbol{x}}(t_{k-N}),\ldots,\hat{\boldsymbol{x}}(t_k)]^T$ throughout the estimation horizon and the recent parameter estimate $\hat{\alpha}(t_{k-N})$.

Update $\hat{\boldsymbol{x}}$ with the last element of the obtained state estimate trajectory and $\hat{\alpha}$, and send them to the NMPC.

Update the weighting matrix $\boldsymbol{R}$ (26b) and dynamics model (2) with the current parameter estimate $\hat{\alpha}$.

Solve the NMPC (24)

Obtain control input trajectory $\mathscr{U} = [\boldsymbol{\tau}(t_k),\boldsymbol{\tau}(t_{k+1}),\ldots,\boldsymbol{\tau}(t_{k+N-1})]^T$ throughout the prediction horizon.

Update $\boldsymbol{\tau}$ with the first element of the control input trajectory, and send forces and torque to the Coordination.

Send the forces to the thusters f_i, $i = 1,\ldots,4n$.

Shift forward the estimation and prediction horizons.

Wait for a new measurement $\boldsymbol{y}_m$.

end while

D. Real-Time Iteration Scheme

The direct multiple shooting method is incorporated with a generalized Gauss-Newton method that is derived from the classical Newton method for least-squares problems. The superiority of this method is that it does not require arduous computations of second derivatives. On the other hand, since it is an iterative method, it is difficult to decide the number of iterations to achieve a desired accuracy. To cope with this challenge, a single iteration is employed for having quick feedback time, and the initial value of each optimization problem takes on the value of the previous one intelligently [22]. As a consequence, this solution improves the convergence of the Gauss-Newton method. Furthermore, qpOASES software package for online active set strategy as a QP solver is used in this study [23].

As can be seen from the NMPC and NMHE formulations respectively in (24) and (19), they are similar problems. The arrival cost in NMHE, i.e., an approximation of the infinite horizon cost before estimation horizon, is the counterpart of the terminal penalty in NMPC, i.e., an approximation of the infinite horizon cost after the prediction horizon. In quadratic costs, differences between the target trajectory and system response are summed in NMPC, while differences between the measurements and measurement function are summed in NMHE. Therefore, the fusion of the multiple shooting method and Gauss-Newton method is employed for NMHE.

In this paper, the Gauss-Newton iteration is splitted into

Algorithm 2 Real-time iteration scheme

Initialize:

$\boldsymbol{x}(t_k) = \boldsymbol{x}(0)$, $\hat{\boldsymbol{x}}(t_k) = \hat{\boldsymbol{x}}(0)$ and $\boldsymbol{\tau}(t_k) = \boldsymbol{\tau}(0)$

Repeat online:

1. Preparation step
   1.1. Shift the previous solution.
   1.2. Evaluate objective and constraints in (19) for NMHE and in (24) for NMPC.
   1.3. Generate corresponding sensitives.
   1.4. Wait for new measurements for NMHE and new estimates for NMPC.
2. Feedback step
   2.1. Compute linear term in QP.
   2.2. Solve the sparse QP.
   2.3. Send the updated state and parameter estimates to NMPC, and control signal to real-time system.
   2.4. Update Nonlinear Programming (NLP) variables for NMHE and NMPC.

two parts: preparation and feedback steps [24], [25]. The preparation step is executed after the feedback step, and the feedback step is executed after measurements for NMHE and estimates for NMPC are available. In the preparation step, the system dynamics are integrated with the previous solution, and objectives, constraints and corresponding sensitives are evaluated. In the feedback step, a single quadratic programming is solved with current estimates for the NMPC and current measurements for the NMHE. Thus, new estimates for the NMHE and new control signals for the NMPC are computed. The proposed method minimizes feedback delay and produces similar results with higher computational efficiency when compared to the classical method. The real-time iteration scheme for NMPC and NMHE is summarized in Algorithm 2.

IV. Experiments

A. Roboats: Robotic Boats

The prototype Roboat used in this study is 0.9 m long $\times$ 0.45 m wide $\times$ 0.15 m height and approximately an $1:4$ scale of a full-size Roboat, which is 4 m long $\times$ 2 m wide. The prototype Roboat is powered by Lithium Ion batteries that offer up to 2 h duration. An indoor positioning system (Marvelmind robotics, USA) is used to acquire centimeter-level accurate positional information in the swimming pool. In addition to positional information, two antennas are mounted straight up the center of the Roboat to obtain the heading angle information. This indoor positioning system provides approximately positional information with an accuracy of 5 cm and heading angle information with an accuracy of 0.07 rad. Moreover, an inertial measurement unit (LORD Microstain, 3DM-GX5-25, USA), which provides angular rate information with an accuracy of 0.01 rad, is mounted on the Roboat body's principal axes. There are four T200 thrusters (BlueRobotics, USA), each of which is capable of providing 35 N force, and four motor controllers, which are one channel motor drivers that use digital control signals to drive thrusters per channel.

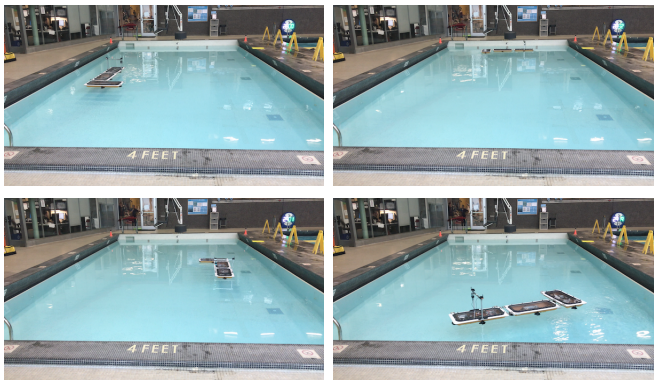


Fig. 4. Roboats with an L-shape configuration in the swimming pool, MIT, Cambridge, MA, USA.

Algorithm is executed on an onboard computer (2.5GHz, 64bit, dual-core CPU with 16 GB of memory) and updated at a rate of 5 Hz.The Arduino Mega 2560, which is a microcontroller board, is used as an a general-purpose input/output for each Roboat and radio frequency modules (Digi XBee S1, USA) are used for communications between multiple Roboats.

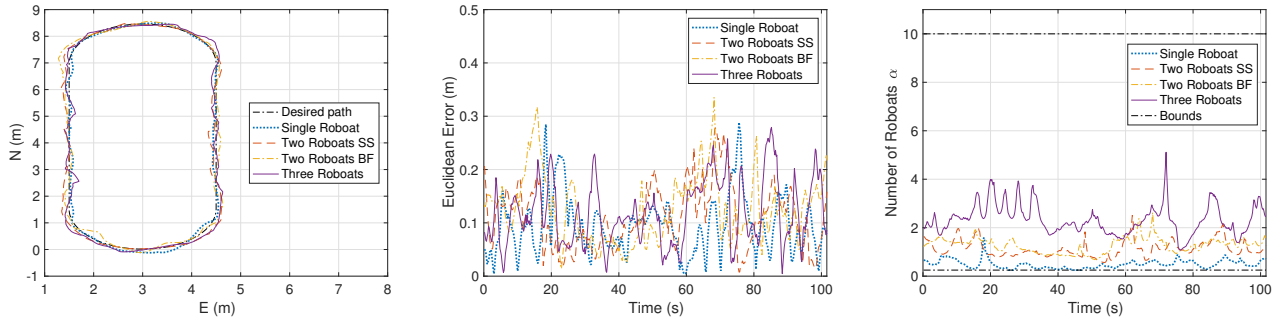
B. Experimental Results

The identified parameters for Roboat are found as $m_{11} = 21.67$ kg, $m_{22} = 39.08$ kg, $m_{33} = 14.56$ kg m^2, $\zeta_u = 23.52$ kg s^{-1}, $\zeta_v = 22.32$ kg s^{-1} and $\zeta_r = 3.762$ kg m^2 s^{-1}. The sampling time is set to 0.2 s, while the prediction horizon in NMPC, as well as the estimation horizon in MHE, is set to 3 s (e.g., $N = 15$) in this study. The experiments have been carried out in a 12 m $\times$ 6 m swimming pool (See Fig. 4).

Throughout the experiments, the developed LB-NMPC algorithm for Roboats is evaluated for a desired path consisting of two straight lines and two curves. In Fig. 5(a), path-tracking performances of different number of Roboats with different configurations are shown. The developed algorithm gives similar path-taking performance for each configuration.

The Euclidean errors for different configurations of Roboats are shown in Fig. 5(b). The mean values of the Euclidean errors are respectively 0.0950 m, 0.1115 m, 0.1259 m and 0.1191 m for a single Roboat, two Roboats with side-by-side (SS) configuration, two Roboats with back-to-front (BF) configuration and three Roboats with an L-shape configuration. The LB-NMPC algorithm gives similar Euclidean errors for different shape and number of connected Roboats. This show the capability of the proposed algorithm in this paper.

The estimated values for the parameter α for different configurations of Roboats are shown in Fig. 5(c). The parameter α increases, while the number of Roboats increases as expected. Since the estimation horizon is set to 3 s in the MHE, such a short estimation horizon is enough to capture the steady-state response of a system. If we also want to capture the transient response of a system, the estimation horizon must be long enough; however, in this case, the estimate for the parameter α cannot be a recent value. The reason is that the parameter is assumed to be time-invariant

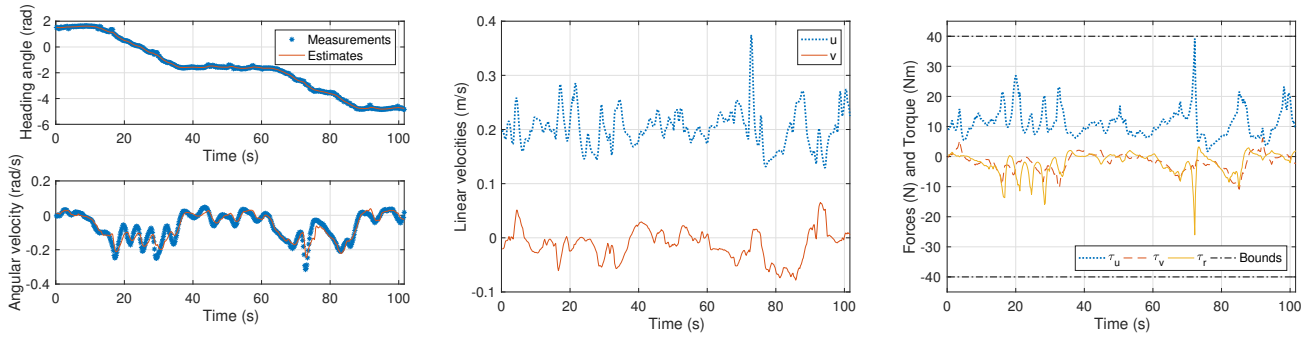


(a) Desired and actual paths. LB-NMPC algorithm gives similar path-tracking performances for different configurations.

(b) Euclidian errors. The mean values of tracking errors are around 12 cm for different configurations.

(c) Estimates for the number of boats. The estimates stay within the lower and upper bounds.

Fig. 5. Experimental results for a single Roboat, two Roboats with side-by-side (SS) configuration, two Roboats with back-to-front (BF) configuration and three Roboats with an L-shape configuration.



(a) Measurements and estimates of the heading angle and angular velocity.

(b) Estimates of the linear velocities: longitudinal velocity u and lateral velocity v.

(c) Control inputs generated by the NMPC: the forces τ_u, τ_v and the torque τ_r. They stay within the lower and upper bounds.

Fig. 6. Experimental results for three Roboats with an L-shape configuration.

throughout the estimation horizon, while it is time-varying in the arrival cost in (19a), (20a) and (20b). Therefore, we can capture only the steady-state behavior of connected Roboats in this study, which explains the variations on the estimated values for the parameter α. Furthermore, the estimates stay within the lower and upper bounds specified in (19e) in Section III-B, which shows the capability of MHE method.

The measured and estimated heading angle and angular velocity for three Roboats with an L-shape configuration are shown in Fig. 6(a). This demonstrates that the MHE can successfully deal with noise on the measurements. The measurements and estimates for the heading angle are constant while the system tracks the straight lines. They decrease while the system tracks curved lines since the system makes right turns. Moreover, the coefficient for the heading angle is penalized less than the coefficients for the positions in (26a). The reason is that the purpose of the LB-NMPC algorithm is to track a desired path accurately; therefore, we penalize the heading angle to find the direction of Roboats and to avoid oscillations around the desired path. The reference angular velocity is set to 0 and -0.13 rad s^{-1} for straight and curved lines, respectively. Therefore, the measurements and estimates for the angular velocity change around these reference values throughout the experiments as the system is on-track.

The estimates for the linear velocities (e.g., longitudinal velocity u and lateral velocity v) for three Roboats with an L-shape configuration are shown in Fig. 6(b). The reference linear velocities are respectively set to 0.2 m s^{-1} and 0 m s^{-1} for the longitudinal velocity u and lateral velocity v throughout the desired path generation. The estimates change around the these reference values as the system is on-track. If the system was not on-track, these estimates would be different than the reference values because the system would try to reach the point on the desired path. Moreover, since the purpose of the LB-NMPC algorithm for connected-Roboats is to obtain high-accurate path-tracking performance, we do not need to penalize the deviations in the linear velocities in the cost function of the NMPC in (26a). They could be penalized to obtain more stable path-tracking performance to avoid oscillations around the desired path if required.

The control signals (e.g., forces and torque) for three Roboats with an L-shape configuration are shown in Fig. 6(c). The generated control signals by the NMPC stay within the lower and upper bounds specified in (27) in Section III-C. The coefficient for the torque is larger than the coefficients for the forces in the weighing matrix (26b) for the NMPC design. The reason is that the drag (e.g., fluid resistance) force for yaw motion is smaller than the ones for the longitudinal and lateral movements for Roboats.

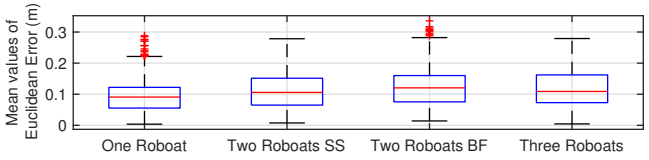


Fig. 7. Mean values of the Euclidean errors for four different configurations.

In order to illustrate that the viability of the proposed LB-NMPC algorithm, the experiments are repeated ten times while the mean values of the Euclidean errors are recorded. A box plot presented in Fig. 7 is prepared for statistical information. It is evident from the figure that the mean values for the Euclidean errors remain the same. This is in accordance with our expectation as the learning helps NMPC to adapt according to the different number and shape of connected-Roboats.

The LB-NMPC algorithm has been run on an onboard computer (2.5 GHz, 64bit, dual-core processor, 16GB RAM) throughout the experiments. The mean value of execution times for LB-NMPC is approximately 0.45 ms. This shows the success of the real-time iteration scheme proposed in this paper.

V. Conclusion

In this paper, a novel Learning-based Nonlinear Model Predictive Control (LB-NMPC) algorithm for reconfigurable autonomous vessels has been elaborated to obtain accurate path-tracking. NMHE estimates the unknown parameter by using on-board sensors and the NMPC designed on the augmented model enables accurate path-tracking performance. The experimental results show that the LB-NMPC algorithm is capable of giving less than 12 cm averagely for any feasible configuration of connected-vessels. Moreover, thanks to a single Gauss-newton iteration principle, the LB-NMPC algorithm is solved within less than a millisecond in real-time.

Acknowledgement

The authors would like to thank to Mr. Pietro Leoni, who is the designer of the Roboat.

Dr. Kayacan was a postdoctoral researcher with the Senseable City Laboratory and Computer Science & Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA, USA when the bulk of this work was undertaken.

References

[1] M. Yim, W. Shen, B. Salemi, D. Rus, M. Moll, H. Lipson, E. Klavins, and G. S. Chirikjian, "Modular self-reconfigurable robot systems," *IEEE Robot. Autom. Mag.*, vol. 14, no. 1, pp. 43–52, March 2007.

[2] S. Murata and H. Kurokawa, "Self-reconfigurable robots," *IEEE Robot. Autom. Mag.*, vol. 14, no. 1, pp. 71–78, March 2007.

[3] L. Furno, M. Blanke, R. Galeazzi, and D. J. Christensen, "Self-reconfiguration of modular underwater robots using an energy heuristic," in *IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, Sept 2017, pp. 6277–6284.

[4] J. Paulos, N. Eckenstein, T. Tosun, J. Seo, J. Davey, J. Greco, V. Kumar, and M. Yim, "Automated self-assembly of large maritime structures by a team of robotic boats," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 3, pp. 958–968, July 2015.

[5] T. I. Fossen and A. M. Lekkas, "Direct and indirect adaptive integral line-of-sight path-following controllers for marine craft exposed to ocean currents," *Int. J. Adapt. Control Signal Process.*, vol. 31, no. 4, pp. 445–463, 2017.

[6] J. Shin, D. J. Kwak, and Y. Lee, "Adaptive path-following control for an unmanned surface vessel using an identified dynamic model," *IEEE/ASME Trans. Mechatronics*, vol. 22, no. 3, pp. 1143–1153, 2017.

[7] S. Dai, M. Wang, and C. Wang, "Neural learning control of marine surface vessels with guaranteed transient tracking performance," *IEEE Trans. Ind. Electron.*, vol. 63, no. 3, pp. 1717–1727, March 2016.

[8] W. He, Z. Yin, and C. Sun, "Adaptive neural network control of a marine vessel with constraints using the asymmetric barrier lyapunov function," *IEEE Trans. Cybern.*, vol. 47, no. 7, pp. 1641–1651, 2017.

[9] R. Oung and R. D'Andrea, "The distributed flight array: Design, implementation, and analysis of a modular vertical take-off and landing vehicle," *Int. J. Robotics Research*, vol. 33, no. 3, pp. 375–400, 2014.

[10] M. C. Nielsen, M. Blanke, and I. Schjolberg, "Efficient modelling methodology for reconfigurable underwater robots," *IFAC-PapersOnLine*, vol. 49, no. 23, pp. 74 – 80, 2016, 10th IFAC Conference on Control Applications in Marine Systems.

[11] S. Park, E. Kayacan, C. Ratti, and D. Rus, "Coordinated control of a reconfigurable multi-vessel platform: Robust control approach," in *2019 IEEE Int. Conf. on Robot. and Autom.*, May 2019, pp. 1–6.

[12] D. Chwa, "Global tracking control of underactuated ships with input and velocity constraints using dynamic surface control method," *IEEE Trans. Control Syst. Technol.*, vol. 19, no. 6, pp. 1357–1370, Nov 2011.

[13] M. Park, S. Chitta, A. Teichman, and M. Yim, "Automatic configuration recognition methods in modular robots," *International Journal of Robotics Research*, vol. 27, no. 3-4, pp. 403–421, 2008.

[14] J. Baca, B. Woosley, P. Dasgupta, and C. A. Nelson, "Configuration discovery of modular self-reconfigurable robots: Real-time, distributed, ir+xbee communication method," *Robotics and Autonomous Systems*, vol. 91, pp. 284 – 298, 2017.

[15] C. V. Rao, J. B. Rawlings, and D. Q. Mayne, "Constrained state estimation for nonlinear discrete-time systems: stability and moving horizon approximations," *IEEE Trans. Autom. Control*, vol. 48, no. 2, pp. 246–258, Feb 2003.

[16] E. L. Haseltine and J. B. Rawlings, "Critical evaluation of extended kalman filtering and moving-horizon estimation," *Industrial & Engineering Chemistry Research*, vol. 44, no. 8, pp. 2451–2460, 2005.

[17] E. Kayacan, S. N. Young, J. M. Peschel, and G. Chowdhary, "High-precision control of tracked field robots in the presence of unknown traction coefficients," *Journal of Field Robotics*, vol. 35, no. 7, pp. 1050–1062, 2018.

[18] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789 – 814, 2000.

[19] E. Kayacan, E. Kayacan, I.-M. Chen, H. Ramon, and W. Saeys, *On the Comparison of Model-Based and Model-Free Controllers in Guidance, Navigation and Control of Agricultural Vehicles*. Cham: Springer International Publishing, 2018, pp. 49–73.

[20] E. Kayacan, Z. Zhang, and G. Chowdhary, "Embedded high precision control and corn stand counting algorithms for an ultra-compact 3d printed field robot," in *Proc. Robot.: Sci. Syst.*, Pittsburgh, Pennsylvania, June 2018.

[21] M. Diehl, H. Bock, J. P. Schloder, R. Findeisen, Z. Nagy, and F. Allgower, "Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations," *Journal of Process Control*, vol. 12, no. 4, pp. 577 – 585, 2002.

[22] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl, "qpoases: a parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, vol. 6, no. 4, pp. 327–363, Dec 2014.

[23] M. Vukov, S. Gros, G. Horn, G. Frison, K. Geebelen, J. Jorgensen, J. Swevers, and M. Diehl, "Real-time nonlinear mpc and mhe for a large-scale mechatronic application," *Control Engineering Practice*, vol. 45, pp. 64 – 78, 2015.

[24] E. Kayacan, W. Saeys, H. Ramon, C. Belta, and J. M. Peschel, "Experimental validation of linear and nonlinear mpc on an articulated unmanned ground vehicle," *IEEE/ASME Trans. Mechatronics*, vol. 23, no. 5, pp. 2023–2030, Oct 2018.