Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 55-13 (2022) 85–90

Distributed MPC for Formation Path-Following of Multi-Vehicle Systems

Josef Matouš * **Damiano Varagnolo** * **Kristin Y. Pettersen** *
Claudio Paliotta **

* *Norwegian University of Science and Technology (NTNU), Trondheim, Norway ({josef.matous,damiano.varagnolo,kristin.y.pettersen}@ntnu.no).*
** *SINTEF Community, Trondheim, Norway (claudio.paliotta@sintef.no).*

Abstract: The paper considers the problem of formation path-following of multiple vehicles and proposes a solution based on combining distributed model predictive control with parametrizations of the trajectories of the vehicles using polynomial splines. Introducing such parametrization leads indeed to two potential benefits: a) reducing the number of optimization variables, and b) enabling enforcing constraints on the vehicles in a computationally efficient way. Moreover, the proposed solution formulates the formation path-following problem as a distributed optimization problem that may then be solved using the alternating direction method of multipliers (ADMM). The paper then analyzes the effectiveness of the proposed method via numerical simulations with surface vehicles and differential drive robots.

Keywords: Formation path-following, Distributed model predictive control (DMPC), Alternating direction method of multipliers (ADMM), B-splines

1. INTRODUCTION

Autonomous vehicles are increasingly used in a variety of applications and environments. Formations of autonomous aerial (Anderson et al., 2008), ground (Bergenhem et al., 2012), surface and underwater (Das et al., 2016) vehicles are used in exploration, monitoring, search and rescue, *etc.* In many applications, the goal is to follow a prescribed path. In the literature, the problem is commonly solved using two types of methods — leader-follower and coordinated path-following (see Das et al. (2016); Anderson et al. (2008) for an overview).

In leader-follower schemes, one vehicle is chosen as the leader of the formation. The leader then follows the path without any further constraints, while the followers adjust their speed and position to maintain the desired formation. The control law for the followers can be based *e.g.,* on backstepping (Cui et al., 2010) or model predictive control (MPC) (Wang et al., 2021). The drawback of leader-follower schemes is the lack of feedback due to the unidirectional communication (*i.e.,* the leader does not adjust its velocity based on the followers).

In coordinated path-following schemes, each vehicle follows its own path. Formation is then achieved by coordinating the motion of the vehicles. This can be done *e.g.,* by a consensus algorithm (Borhaug and Pettersen, 2006) or MPC (Kanjanawanishkul and Zell, 2008). The MPC scheme in Kanjanawanishkul and Zell (2008) is based on sampling. Consequently, any constraints on trajectories or states can only be enforced at discrete time-instances. In other words, we have no control over the behavior of the system between the samples. We can mitigate this issue by decreasing the sampling time. However, by decreasing the sampling time, we increase the number of optimized variables, thus increasing the computational and communication requirements.

In recent years, researchers have focused on computationally tractable MPC schemes. One possibility of reducing the computational requirements is to parametrize the vehicles' trajectories using splines. Saska et al. (2016) have proposed a spline-based path planning MPC algorithm for first-order nonholonomic vehicles. The algorithm solves the point-to-point formation tracking problem with static obstacles. Another spline-based MPC algorithm has been proposed by Van Parys and Pipeleers (2017). This algorithm is applicable to a wider range of systems compared to Saska et al. (2016), and it has been demonstrated on point-to-point and trajectory tracking problems. However, there is, to the best of our knowledge, no reported work on how to apply spline-based MPC to the coordinated path-following problem.

The goal of this paper is thus to propose a spline-based MPC strategy for the coordinated path-following problem, test its suitability for the purpose, and understand which trade-offs characterize this scheme. As we explain in the next section, spline-based MPC imposes some assumptions on the structure of the model of the vehicles. The spline-based MPC scheme can thus be seen as a trade-off between lower computational requirements and more restrictive assumptions on the model.

⋆ This work was partly supported by the Research Council of Norway through project No. 302435 and the Centres of Excellence funding scheme, project No. 223254.

10.1016/j.ifacol.2022.07.240

The remainder of the paper is organized as follows. In Section 2, we present the general assumptions on the model of the vehicles and formally define the formation path-following problem. In Section 3, we propose the distributed spline-based MPC scheme. Section 4 presents two case studies. Finally, Section 5 gives some concluding remarks.

2. PROBLEM DESCRIPTION

In this section, we first introduce the assumptions on the model of the vehicles. Then, we define the objective of formation path-following and pose it as an optimization problem.

2.1 Vehicle Model

Here, we discuss the dynamics of a single agent in the network. Let $\boldsymbol{x} \in \mathbb{R}^{n_{\boldsymbol{x}}}$ be the vector of states. We assume that the state vector includes the position of the agent. Without loss of generality, let the first $n_{\boldsymbol{q}}$ states be the position of the vehicle. We can then define the position vector of the vehicle as

$$\boldsymbol{q} = \left[x_1, \ldots, x_{n_{\boldsymbol{q}}}\right]^{\mathrm{T}}. \tag{1}$$

Since vehicles typically move in either two or three dimensions, we assume that $n_{\boldsymbol{q}} \in \{2, 3\}$. Let $\boldsymbol{u} \in \mathbb{R}^{n_{\boldsymbol{q}}}$ be the vector of control inputs. We assume the dynamics of the vehicle to be given by an ordinary differential equation

$$\dot{\boldsymbol{x}} = \boldsymbol{f}\left(\boldsymbol{x}, \boldsymbol{u}\right). \tag{2}$$

Note that we assume the number of inputs to be equal to $n_{\boldsymbol{q}}$. In cases where this assumption does not hold because the vehicle is overactuated, we need to reduce the number of inputs by introducing a control allocation scheme (see, *e.g.,* Johansen and Fossen (2013)).

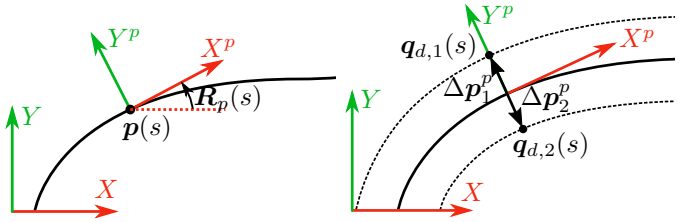
Let $\boldsymbol{y} \in \mathbb{R}^{n_{\boldsymbol{q}}}$ be the output of the system. Note that, in general, the output can be different from the position of the vehicle. However, as discussed in the next paragraph, it must be possible to obtain the position from the output.

We assume the system to be differentially flat (Fliess et al., 1995), *i.e.*, we assume that the input and state can be determined from the output, its derivatives, and antiderivatives. Moreover, we assume that the relation between the output and the position is polynomial. In other words, there exist suitable (nonlinear) functions $\boldsymbol{\phi_u}$ and $\boldsymbol{\phi_x}$, and a multidimensional polynomial function $\boldsymbol{\phi_q}$ of suitable dimensions such that, at any time,

$$\boldsymbol{u} = \boldsymbol{\phi_u}\left(\boldsymbol{y}, \dot{\boldsymbol{y}}, \ddot{\boldsymbol{y}}, \ldots, \boldsymbol{y}^{(r')}\right), \tag{3}$$

$$\boldsymbol{x} = \boldsymbol{\phi_x}\left(\boldsymbol{y}^{(-r'')}, \ldots, \boldsymbol{y}, \dot{\boldsymbol{y}}, \ddot{\boldsymbol{y}}, \ldots, \boldsymbol{y}^{(r')}\right), \tag{4}$$

$$\boldsymbol{q} = \boldsymbol{\phi_q}\left(\boldsymbol{y}^{(-r'')}, \ldots, \boldsymbol{y}, \dot{\boldsymbol{y}}, \ddot{\boldsymbol{y}}, \ldots, \boldsymbol{y}^{(r')}\right), \tag{5}$$



a The inertial coordinate frame (X, Y), and the path-tangential frame (X^p, Y^p).

b Example of a formation. In this example, the relative positions are $\Delta\boldsymbol{p}_1^p = [0, a]^{\mathrm{T}}, \Delta\boldsymbol{p}_2^p = -\Delta\boldsymbol{p}_1^p$, where a is some positive number.

Fig. 1. Illustration of the path-following problem.

where r' and r'' are positive integers.

To model the constraints on the dynamics of the vehicle, we use a multidimensional function $\boldsymbol{e}$. The set of feasible states and inputs is given by

$$\left\{ (\boldsymbol{x}, \boldsymbol{u}) \,\middle|\, \boldsymbol{e}(\boldsymbol{x}, \boldsymbol{u}) \geq \boldsymbol{0} \right\} \tag{6}$$

where the inequality is defined component-wise. We assume that substituting (3), (4) into (6) yields a set of polynomial constraints. In other words, we assume that there exists a multidimensional polynomial function $\boldsymbol{h}$ such that

$$\boldsymbol{e}(\boldsymbol{x}, \boldsymbol{u}) \geq \boldsymbol{0} \iff \boldsymbol{h}\left(\boldsymbol{y}^{(-r'')}, \ldots, \boldsymbol{y}^{(r')}\right) \geq \boldsymbol{0}. \tag{7}$$

2.2 Formation Path-Following

The goal is to control N vehicles, all subject to the dynamics introduced in Section 2.1, so that they move in a prescribed formation while their barycenter follows a given path. Let $\boldsymbol{p} : \mathbb{R} \to \mathbb{R}^{n_{\boldsymbol{q}}}$ be a function that represents this path. We assume that the function is continuously differentiable and its first derivative satisfies

$$\left\| \frac{\partial \boldsymbol{p}(s)}{\partial s} \right\| = 1. \tag{8}$$

This implies that for every path point $\boldsymbol{p}(s)$, we can define a path-tangential coordinate frame and a corresponding rotation matrix $\boldsymbol{R}_p(s)$ between the inertial and path-tangential frames (a 2D example is shown in Fig. 1a).

Let then $\boldsymbol{q}_1(t), \ldots, \boldsymbol{q}_N(t)$ be the trajectories of the N vehicles composing the fleet, and let $\Delta\boldsymbol{p}_1^p, \ldots, \Delta\boldsymbol{p}_N^p$ be the desired positions of the vehicles in the formation, relative to the barycenter. Using this notation, the desired trajectory for agent i is given by

$$\boldsymbol{q}_{d,i}(s) = \boldsymbol{p}(s) + \boldsymbol{R}_p(s)\, \Delta\boldsymbol{p}_i^p, \tag{9}$$

for a given s. An example formation is shown in Fig. 1b.

The objective of the control system is to steer the actual vehicle positions $\boldsymbol{q}_i(t)$ to follow the desired trajectories $\boldsymbol{q}_{d,i}$. Ideally, this means that for a given function $s(t)$ we seek the actual positions to be such that

$$\boldsymbol{q}_i(t) \equiv \boldsymbol{q}_{d,i}\big(s(t)\big), \quad i = 1, \ldots, N. \tag{10}$$

Note that the path parameter $s(t)$ from the previous equation can be treated as an additional *degree of freedom* when designing the controller. Consequently, we also need to find a suitable control law for $s(t)$. For this purpose, let U_d be the desired speed of the barycenter of the formation. When the vehicles follow the path perfectly, the actual speed of the barycenter is given by

$$U(t) = \left\| \dot{\boldsymbol{p}}\big(s(t)\big) \right\| = \left\| \frac{\partial \boldsymbol{p}(s)}{\partial s}\, \dot{s}(t) \right\| = \left| \dot{s}(t) \right|. \tag{11}$$

The equivalence above implies that the path parameter $s(t)$ should thus be chosen such that

$$\dot{s} \equiv U_d. \tag{12}$$

2.3 A centralized solution to the path-formation problem

The problem of finding for each agent i its actuation signal $\boldsymbol{u}_i(t)$ that guarantees following the desired path $\boldsymbol{q}_{d,i}\big(s(t)\big)$ as close as possible can thanks to (3)–(5) be transformed into the problem of finding a corresponding output trajectory $\boldsymbol{y}_i(t)$.

In general, it is not possible to find an output trajectory $\boldsymbol{y}_i(t)$ such that (10) is satisfied, since the dynamics of the

agents are constrained by both (2) and (7). This means that at any time t, there is a position error

$$\widetilde{\boldsymbol{q}}_i(t) = \boldsymbol{q}_i(t) - \boldsymbol{q}_{d,i}\big(s(t)\big). \tag{13}$$

Thanks to (5), we can express $\widetilde{\boldsymbol{q}}_i(t)$ in terms of $\boldsymbol{y}_i(t)$

$$\widetilde{\boldsymbol{q}}_i(t) = \boldsymbol{\phi}_{\boldsymbol{q}}\left(\boldsymbol{y}^{(-r'')}(t), \ldots, \boldsymbol{y}^{(r')}(t)\right) - \boldsymbol{q}_{d,i}\big(s(t)\big), \tag{14}$$

and thus solve the problem by optimizing $\boldsymbol{y}_i(t)$ and $s(t)$.

The problem should be cast in a receding horizon fashion to reject potential disturbances as the mission proceeds. We thus propose to formulate the *centralized* problem of optimizing a part of the trajectory, i.e., $\boldsymbol{y}_i(t:t+T)$, $s(t:t+T)$, as that of optimizing the constrained problem

$$\begin{aligned}\underset{\{\boldsymbol{y}_i(t:t+T)\},s(t:t+T)}{\text{minimize}} \sum_{i=1}^{N} \int_t^{t+T} \widetilde{\boldsymbol{q}}_i^{\mathrm{T}}(\tau)\, \boldsymbol{Q}_{\boldsymbol{p}}\, \widetilde{\boldsymbol{q}}_i(\tau)\mathrm{d}\tau \\ + \int_t^{t+T} Q_s\, \left(\dot{s}(\tau) - U_d\right)^2 \mathrm{d}\tau,\end{aligned} \tag{15}$$

with T being the prediction horizon, $\boldsymbol{Q}_{\boldsymbol{p}}$ and Q_s positive weight matrices, $\widetilde{q}_i$ the position error as defined in (14), and subject to, for every agent $i = 1, \ldots, N$, to the constraints C1 to C3 below:

- C1 the implicit constraint on the inputs and states, *i.e.*, $\boldsymbol{h}\left(\boldsymbol{y}_i^{(-r'')}(\tau), \ldots, \boldsymbol{y}_i^{(r')}(\tau)\right) \geq \boldsymbol{0}, \quad \forall \tau \in [t, t+T]$,
- C2 the constraint on the initial condition of the state of the system, i.e., $\boldsymbol{\phi}_{\boldsymbol{x}}\left(\boldsymbol{y}_i^{(-r'')}(t), \ldots, \boldsymbol{y}_i^{(r')}(t)\right) = \boldsymbol{x}_i(t)$,
- C3 the constraint on the initial condition of the path of the agents, *i.e.*, $s(t)$. In other words, $s(t)$ is not a decision variable, while $s(t+\tau)$ for any $\tau > 0$ is.

We note that the variational problem above may not be solvable using off-the-shelf hardware with limited computing power. For this reason, it will be rewritten below.

2.4 A distributed solution to the path-formation problem

Before doing this rewriting, we note that it is possible to make (15) distributed by letting the path parameter $s(t)$ be a local variable (*i.e.*, $s_i(t)$), and adding a synchronization constraint on the set of $s_i(t)$'s. This leads to the local reformulation

$$\begin{aligned}\underset{\{\boldsymbol{y}_i(t:t+T), s_i(t:t+T)\}}{\text{minimize}} \int_t^{t+T} \widetilde{\boldsymbol{q}}_i^{\mathrm{T}}(\tau)\, \boldsymbol{Q}_{\boldsymbol{p}}\, \widetilde{\boldsymbol{q}}_i(\tau)\mathrm{d}\tau \\ + \int_t^{t+T} Q_s\, \left(\dot{s}_i(\tau) - U_d\right)^2 \mathrm{d}\tau,\end{aligned} \tag{16a}$$

$$\text{subject to } h\left(\boldsymbol{y}_i^{(-r'')}(\tau), \ldots, \boldsymbol{y}_i^{(r')}(\tau)\right) \geq \boldsymbol{0}, \tag{16b}$$

$$\phi_{\boldsymbol{x}}\left(\boldsymbol{y}_i^{(-r'')}(t), \ldots, \boldsymbol{y}_i^{(r')}(t)\right) = \boldsymbol{x}_i(t), \tag{16c}$$

$$s_i(\tau) = s_j(\tau), \quad \forall j, \tau \in [t, t+T] \tag{16d}$$

where j is the index of the generic neighbor of agent i. This formulation is again only an intermediate step towards the approach proposed in this paper, as explained below.

3. A DISTRIBUTED SPLINE-BASED MPC APPROACH TO THE PATH-FORMATION PROBLEM

The goal of this section is to show how constraining $\boldsymbol{y}_i$ and s_i to be splines enables rewriting the variational problem above in a way that is computationally tractable.

For the sake of readability, we will use the convention for which sans-serif fonts (*e.g.*, $\mathbf{y}$) indicate quantities relative to splines, while serif fonts (*e.g.*, $\boldsymbol{y}$) indicate trajectories parametrized in time as above.

3.1 Spline parametrization

Let $\mathbf{b} = [\mathsf{b}_1, \ldots, \mathsf{b}_n]^{\mathrm{T}}$ be the vector of basis functions of a B-spline, and let $\mathbf{y}_i = \left[\mathbf{y}_{i,1}^{\mathrm{T}}, \ldots, \mathbf{y}_{i,n}^{\mathrm{T}}\right]^{\mathrm{T}}$ be a generic matrix and $\mathbf{s}_i = [\mathsf{s}_{i,1}, \ldots, \mathsf{s}_{i,n}]^{\mathrm{T}}$ a generic vector of spline coefficients. Assume then that the trajectories and path parameters may be expressed as B-splines, i.e., as

$$\boldsymbol{y}_i(\tau) = \sum_{l=1}^{n} \mathbf{y}_{i,l}\, \mathsf{b}_l(\tau) = \mathbf{y}_i^{\mathrm{T}}\, \mathbf{b}(\tau), \tag{17}$$

$$s_i(\tau) = \sum_{l=1}^{n} \mathsf{s}_{i,l}\, \mathsf{b}_l(\tau) = \mathbf{s}_i^{\mathrm{T}}\, \mathbf{b}(\tau). \tag{18}$$

This assumption implies the possibility of exploiting the convex hull property

$$\mathbf{y}_i \geq \mathbf{0} \implies \boldsymbol{y}(\tau) \geq \mathbf{0}, \tag{19}$$

that implies that any polynomial constraint on a spline can be replaced by a (stricter) constraint on the spline coefficients. In other words, by assuming the output to be a spline, we assume that there exists a function h such that

$$\mathsf{h}\left(\mathbf{y}_i\right) \geq \mathbf{0} \implies h\left(\boldsymbol{y}_i^{(-r'')}(\tau), \ldots, \boldsymbol{y}_i^{(r')}(\tau)\right) \geq \mathbf{0}. \tag{20}$$

This eventually enables us to rewrite the trajectory optimization problems in Section 2 as corresponding spline-based MPC problems.

To do so, each agent i must locally approximate the path function and the associated rotation matrix as polynomials

$$\boldsymbol{p}(s) \approx \boldsymbol{p}_0 + \boldsymbol{p}_1\, s + \ldots + \boldsymbol{p}_m\, s^m, \tag{21}$$

$$\boldsymbol{R}_p(s) \approx \boldsymbol{R}_{p,0} + \boldsymbol{R}_{p,1}\, s + \ldots + \boldsymbol{R}_{p,m}\, s^m, \tag{22}$$

over an interval $[s_i(t), s_i(t) + s_T]$, where t is the current time and s_T is chosen such that $s_T \geq U_d T$. We then need to impose an additional constraint on the path parameter

$$s_i(t) \leq s_i(\tau) \leq s_i(t) + s_T, \quad \forall \tau \in [t, t+T], \tag{23}$$

to ensure that the polynomial approximation is valid.

This approximation transforms the criterion from (16a) into a polynomial function. The optimization problem (16) can then be reformulated in terms of spline coefficients

$$\underset{\mathbf{y}_i, \mathbf{s}_i}{\text{minimize}}\, J_i\left(\mathbf{y}_i, \mathbf{s}_i\right), \tag{24a}$$

$$\text{subject to } \mathbf{y}_i \in \mathsf{Y}_i, \tag{24b}$$

$$\mathbf{s}_i \in \mathsf{S}_i, \tag{24c}$$

$$\mathbf{s}_i = \mathbf{s}_j, \quad \forall j = 1, \ldots, N, \tag{24d}$$

where J_i is the objective function from (16a), reformulated using the spline coefficients, and Y_i and S_i are the sets of feasible coefficients given by (16b), (16c) and (23).

This optimization problem is then solved in discrete time-steps. Similarly to collocation-based MPC, we can use the results from the previous time-step to "warm-start" the optimization problem. We do this by extrapolating the previous results over the new horizon (see Fig. 2).

3.2 ADMM algorithm

In Boyd et al. (2011), it is discussed that ADMM tends to converge to "modest accuracy" within a few iterations. Due to this property, ADMM is often used to solve distributed MPC problems.

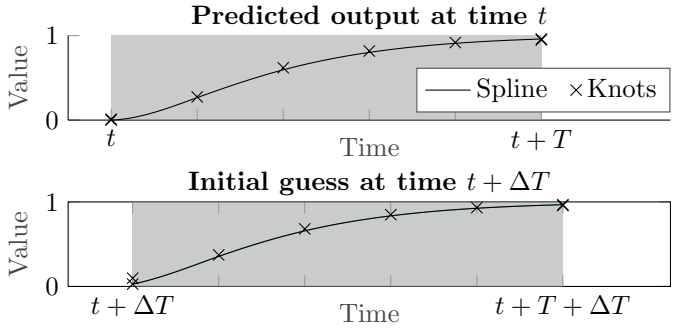


Fig. 2. Warm-starting the optimization problem. The grey area represents the prediction horizon.

We assume *synchronous bidirectional reliable* communication. In other words, we assume that all vehicles exchange information simultaneously and there are no packet losses. Bidirectional communication implies that the communication network can be described by an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, 2, \ldots, N\}$ correspond to the agents, and $\mathcal{E} \subseteq \mathcal{V}^2$ represents the communication between pairs of agents, *i.e.,* $(i, j) \in \mathcal{E}$ implies that agent i can exchange information with agent j. We further assume that $\mathcal{G}$ is connected.

We use relaxed ADMM (Bastianello et al., 2021) to solve the problem. Let $\mathcal{N}_i$ be the set of neighbors of agent i. Relaxed ADMM solves the optimization problem (24) by introducing an auxiliary variable $\mathbf{z}_{ji}$ for all $i \in \mathcal{V}, j \in \mathcal{N}_i$. The optimization problem is then solved iteratively in two steps. First, we compute $\mathbf{y}_i$ and $\mathbf{s}_i$ by solving

$$\mathbf{y}_i, \mathbf{s}_i \leftarrow \underset{\mathbf{y}_i \in \mathsf{Y}_i, \mathbf{s}_i \in \mathsf{S}_i}{\arg\min}\ \mathcal{L}_i\left(\mathbf{y}_i, \mathbf{s}_i, \mathbf{z}_{ji}\right), \tag{25}$$

where $\mathcal{L}_i(\mathbf{y}_i, \mathbf{s}_i, \mathbf{z}_{ji})$ is the augmented Lagrangian given by

$$\mathcal{L}_i(\mathbf{y}_i, \mathbf{s}_i, \mathbf{z}_{ji}) = J_i(\mathbf{y}_i, \mathbf{s}_i) - \sum_{j \in \mathcal{N}_i} \mathbf{z}_{ji}^{\mathrm{T}} \mathbf{s}_i + \frac{\rho}{2} d_i \left\|\mathbf{s}_i\right\|^2, \tag{26}$$

where $\rho > 0$ is a penalty and d_i is the cardinality of $\mathcal{N}_i$. In the second step, we update the auxiliary variables

$$\mathbf{z}_{ji} \leftarrow (1-\alpha)\mathbf{z}_{ji} + \alpha\left(2\rho \mathbf{s}_j - \mathbf{z}_{ij}\right), \tag{27}$$

where $0 < \alpha < 1$ is the step size. To perform this step, each agent $j \in \mathcal{N}_i$ sends a packet

$$\mathbf{w}_{ji} = 2\rho \mathbf{s}_i - \mathbf{z}_{ji}, \tag{28}$$

to agent i. The update law (27) then becomes

$$\mathbf{z}_{ji} \leftarrow (1-\alpha)\mathbf{z}_{ji} + \alpha \mathbf{w}_{ji}. \tag{29}$$

To further reduce the needed communication bandwidth, we only perform one ADMM iteration per MPC step. An overview of the resulting distributed MPC is shown in Algorithm 1.

Algorithm 1 ADMM for Distributed MPC

1: **Initialization:** Perform several ADMM iterations to get $\mathbf{y}_i^0, \mathbf{s}_i^0$ and $\mathbf{z}_{ij}^0$
2: **for** $k = 1, 2, \ldots$ **do** every ΔT
3: Use extrapolation to estimate $\widehat{\mathbf{y}}_i^k, \widehat{\mathbf{s}}_i^k$ and $\widehat{\mathbf{z}}_{ij}^k$
4: Perform one ADMM iteration using $\widehat{\mathbf{y}}_i^k, \widehat{\mathbf{s}}_i^k$ and $\widehat{\mathbf{z}}_{ij}^k$ as the initial guess
5: **end for**

4. CASE STUDIES

In this section, we demonstrate the proposed MPC scheme on marine vehicles and differential drive robots. In both cases, the goal is to follow a sine-wave path given by

$$\boldsymbol{p}(s) = \left[\sigma(s),\ 15 \sin\left(\frac{\pi}{100}\,\sigma(s)\right)\right]^{\mathrm{T}}, \tag{30}$$

where $\sigma(s)$ is a function chosen such that (8) is satisfied and $\sigma(0) = 0$. This path is then locally approximated using a fifth-order polynomial with $s_T = 75$.

In both case studies, the prediction horizon is set to 50 seconds. The path parameter and outputs are represented by cubic splines with 11 breakpoints. Consequently, each spline is represented by 13 coefficients.

In both case studies, we consider a formation of six vehicles. The formation is chosen such that the vehicles form an equilateral triangle with a side length of 20 meters. The desired relative positions of the vehicles, as well as the communication graph are illustrated in Fig. 3c.

4.1 Marine vehicles

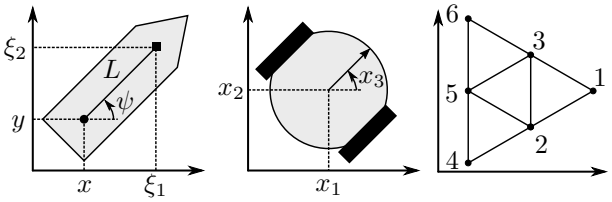
In the first case study, we consider marine vehicles with three degrees of freedom. The model presented in this section describes both autonomous surface vehicles (ASVs) and autonomous underwater vehicles (AUVs) moving in the horizontal plane. First, let us present the assumptions this model is based on.

- A1 The vehicles are port-starboard symmetric.
- A2 The vehicles are maneuvering at low speeds. Consequently, the hydrodynamic damping is linear.
- A3 The vehicles are affected by a constant irrotational ocean current. Its velocity in the inertial coordinate frame is given by $\boldsymbol{V} = [V_x, V_y]^{\mathrm{T}}$.

Under these assumptions, the model of the vehicle is given by the following differential equations (Fossen, 2011)

$$\dot{x} = u_r \cos\psi - v_r \sin\psi + V_x \tag{31a}$$
$$\dot{y} = u_r \sin\psi + v_r \cos\psi + V_y \tag{31b}$$
$$\dot{\psi} = r \tag{31c}$$
$$\dot{u}_r = F_{u_r}(v_r) + \tau_u \tag{31d}$$
$$\dot{v}_r = X(u_r) r + Y(u_r) v_r \tag{31e}$$
$$\dot{r} = F_r(u_r, v_r, r) + \tau_r \tag{31f}$$

where $\eta \triangleq [x, y, \psi]^{\mathrm{T}}$ is the pose of the vehicle in the *North-East-Down* (NED) coordinate frame, $\nu_r \triangleq [u_r, v_r, r]^{\mathrm{T}}$ are the relative (with respect to the ocean current) velocities in the body-fixed coordinate frame, namely the surge velocity, sway velocity, and yaw rate, respectively, and τ_u, τ_r are the control inputs. The functions F_{u_r}, F_r, X and Y represent the Coriolis, centrifugal, and hydrodynamic effects. Due to space constraints, the definitions of these functions are omitted. For more details, the reader is referred to Fossen (2011); Borhaug et al. (2007).



a Marine vehicle b Differential drive robot c Desired formation

Fig. 3. Graphical representations of the vehicles and the formation used in the case studies.

Because the vehicle is underactuated (second-order nonholonomic), we cannot use the origin of the body-fixed frame (x, y) as the output of our system. Instead, we choose a different point, located on the central axis (x-axis) of the vehicle, as the output of our system (see Fig. 3a). In the literature, this point is referred to as the *hand position* of the vehicle (Lawton et al., 2003; Paliotta et al., 2019). The hand position is defined as

$$\boldsymbol{y} = [x + L\,\cos(\psi), y + L\,\sin(\psi)]^{\mathrm{T}}, \tag{32}$$

where $L > 0$ is the hand length.

The advantage of using the hand position is that we can employ output-feedback linearization to simplify the dynamics. This procedure follows along the lines of Paliotta et al. (2019) with one modification. In Paliotta et al. (2019), the ocean current is assumed to be unknown. Here, we assume that the vehicle either can measure the ocean current or employs an observer that provides an accurate estimate of the current (see, *e.g.,* Zhu et al. (2016)).

We thus introduce the following change of coordinates

$$\xi_1 = x + L\,\cos\psi, \tag{33a}$$
$$\xi_2 = y + L\,\sin\psi, \tag{33b}$$
$$\xi_3 = u_r\cos\psi - v_r\sin\psi + V_x - Lr\,\cos\psi, \tag{33c}$$
$$\xi_4 = u_r\sin\psi + v_r\cos\psi + V_y + Lr\,\sin\psi, \tag{33d}$$

and the following change of inputs

$$\begin{bmatrix}\tau_u \\ L\tau_r\end{bmatrix} = \mathfrak{R}(\psi)^{\mathrm{T}}\begin{bmatrix}-F_{\xi_3}(\psi, u_r, v_r, r) + u_1 \\ -F_{\xi_4}(\psi, u_r, v_r, r) + u_2\end{bmatrix}, \tag{34}$$

where $\mathfrak{R} : \mathbb{R} \mapsto \mathrm{SO}(2)$ denotes the rotation matrix and

$$\begin{bmatrix}F_{\xi_3}(\cdot) \\ F_{\xi_4}(\cdot)\end{bmatrix} = \mathfrak{R}(\psi)\begin{bmatrix}F_{u_r}(\cdot) - v_r r - Lr^2 \\ u_r r + X(\cdot)r + Y(\cdot)v_r + F_r(\cdot)L\end{bmatrix}. \tag{35}$$

Using output-feedback linearization, we have simplified the system to a double integrator

$$\ddot{\boldsymbol{y}} = \boldsymbol{u}. \tag{36}$$

Having transformed the system model into the required form, we validated the proposed method in numerical simulations. The simulations were carried out on a 3DOF model of the light autonomous underwater vehicle (LAUV) developed at the University of Porto (Sousa et al., 2012). The parameters are shown in Table 1a. The results are shown in Fig. 4. The left plot shows how the vehicles converge to the desired formation. The right plot shows the path-following errors $\widetilde{\boldsymbol{q}}_i$. We only show the first 50 seconds since the errors converge to zero afterwards.

4.2 Differential drive robots

In the second case study, we consider differential drive robots modeled as unicycles. The model is given by

Table 1. Simulation parameters

a Marine vehicles

Parameter	Value
ΔT	1
$\boldsymbol{Q}_p$	$\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$
Q_s	10
ρ	10
α	0.6
L	1
$\boldsymbol{V}$	$\begin{bmatrix}0.15\\ 0.1\end{bmatrix}$

b Differential drive robots

Parameter	Value
ΔT	0.1
$\boldsymbol{Q}_p$	$\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$
Q_s	10
ρ	10
α	0.6
$u_{1,\min}$	-1
$u_{1,\max}$	2
$u_{2,\min}$	$-\pi/8$
$u_{2,\max}$	$\pi/8$

$$\dot{x}_1 = u_1\,\cos x_3, \tag{37a}$$
$$\dot{x}_2 = u_1\,\sin x_3, \tag{37b}$$
$$\dot{x}_3 = u_2, \tag{37c}$$

where x_1, x_2 give the position, x_3 is the orientation of the vehicle, and u_1 and u_2 are the tangential and angular velocities.

Similarly to the previous case, we could use the hand position to enable the application of the spline-based MPC. However, doing so would prevent us from imposing constraints on the inputs. Instead, we will use the procedure from Van Parys and Pipeleers (2017). First, we introduce $z = \tan\frac{x_3}{2}$ and use the following trigonometric identities

$$\cos x_3 = \frac{1 - z^2}{1 + z^2}, \qquad \sin x_3 = \frac{2z}{1 + z^2}. \tag{38}$$

Next, we substitute z and a modified input $\bar{u}_1 = \frac{u_1}{1+z^2}$ into the first two lines of (37) to obtain

$$\dot{x}_1 = \bar{u}_1\left(1 - z^2\right), \qquad \dot{x}_2 = 2\bar{u}_1 z. \tag{39}$$

We choose $\boldsymbol{y} = [\bar{u}_1, z]^{\mathrm{T}}$ as the output of the system. The states and inputs can then be expressed as

$$x_1(t) = \int_0^t y_1(\tau)\left(1 - y_2^2(\tau)\right)\,\mathrm{d}\tau + x_1(0), \tag{40a}$$
$$x_2(t) = \int_0^t 2y_1(\tau)y_2(\tau)\,\mathrm{d}\tau + x_2(0), \tag{40b}$$
$$x_3(t) = 2\arctan y_2(t), \tag{40c}$$
$$u_1(t) = y_1(t)\left(1 + y_2^2(t)\right), \tag{40d}$$
$$u_2(t) = \frac{2\dot{y}_2(t)}{1 + y_2^2(t)}. \tag{40e}$$

Let us assume that there are no constraints on the states and the constraints on the inputs are given by

$$u_{1,\min} \le u_1(t) \le u_{1,\max}, \quad u_{2,\min} \le u_2(t) \le u_{2,\max}, \tag{41}$$

From (40d), (40e), the constraints can be expressed as

$$u_{1,\min} \le y_1(t)\left(1 + y_2^2(t)\right) \le u_{1,\max}, \tag{42a}$$
$$u_{2,\min}\left(1 + y_2^2(t)\right) \le 2\dot{y}_2(t) \le u_{2,\max}\left(1 + y_2^2(t)\right). \tag{42b}$$

We have thus shown how to express the states, inputs and constraints in terms of the outputs.

The results of numerical simulations are shown in Fig. 5. The parameters are shown in Table 1b. Due to the numerical inaccuracies caused by (40) and arising primarily from the multiplication and division of splines, the MPC timestep ΔT must be shorter than in the previous case-study.

5. CONCLUSIONS

In this paper, we have proposed a distributed spline-based MPC scheme for the formation path-following problem. We have shown that using splines makes the distributed control problem computationally tractable. Compared to collocation, the spline parametrization allows us to represent a longer prediction horizon using fewer variables. This is also beneficial for the communication, and thus makes it easier to do distributed control in environments where the communication bandwidth is limited (*e.g.,* underwater).

One might argue that restricting the output to splines limits the subspace of feasible trajectories. However, simulation results show that cubic splines provide a good approximation of many curves. Another limiting factor is the need for differential flatness. However, it is often possible to

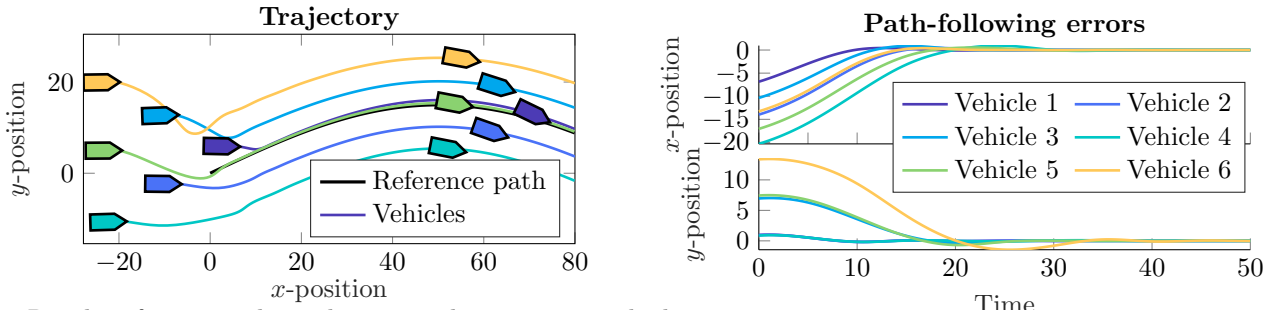


Fig. 4. Results of numerical simulations with six marine vehicles.

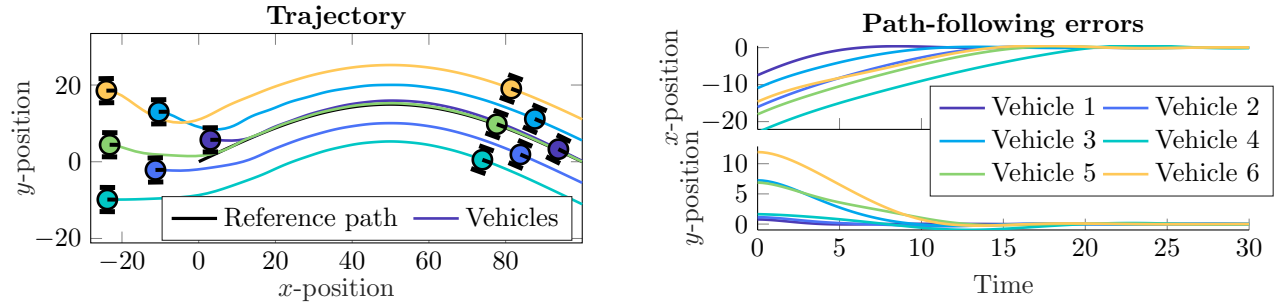


Fig. 5. Results of numerical simulations with six differential drive robots.

simplify the structure of the model to guarantee differential flatness. The proposed spline-based MPC scheme can thus be seen as a trade-off between lower computational requirements and more restrictive assumptions on the model.

REFERENCES

Anderson, B., Fidan, B., Yu, C., and Walle, D. (2008). UAV formation control: Theory and application. In *Recent Advances in Learning and Control. Lecture Notes in Control and Information Sciences*. Springer.

Bastianello, N., Carli, R., Schenato, L., and Todescato, M. (2021). Asynchronous distributed optimization over lossy networks via relaxed ADMM: Stability and linear convergence. *IEEE Transactions on Automatic Control*.

Bergenhem, C., Shladover, S., Coelingh, E., Englund, C., and Tsugawa, S. (2012). Overview of platooning systems. In *Proc. 19th ITS World Congress*.

Borhaug, E., Pavlov, A., and Pettersen, K.Y. (2007). Straight line path following for formations of underactuated underwater vehicles. In *Proc. 46th IEEE Conference on Decision and Control*, 2905–2912.

Borhaug, E. and Pettersen, K.Y. (2006). Formation control of 6-DOF Euler-Lagrange systems with restricted inter-vehicle communication. In *Proc. 45th IEEE Conference on Decision and Control*, 5718–5723.

Boyd, S., Parikh, N., and Chu, E. (2011). *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Foundations and Trends in Machine Learning.

Cui, R., Sam Ge, S., Voon Ee How, B., and Sang Choo, Y. (2010). Leader–follower formation control of underactuated autonomous underwater vehicles. *Ocean Engineering*, 37(17), 1491–1502.

Das, B., Subudhi, B., and Pati, B.B. (2016). Cooperative formation control of autonomous underwater vehicles: An overview. *International Journal of Automation and Computing*, 13(3), 199–225.

Fliess, M., Lévine, J., Martin, P., and Rouchon, P. (1995). Flatness and defect of non-linear systems: introductory theory and examples. *International Journal of Control*.

Fossen, T.I. (2011). *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley & Sons.

Johansen, T.A. and Fossen, T.I. (2013). Control allocation — A survey. *Automatica*, 49(5), 1087–1103.

Kanjanawanishkul, K. and Zell, A. (2008). Distributed model predictive control for coordinated path following control of omnidirectional mobile robots. In *Proc. 2008 IEEE International Conference on Systems, Man and Cybernetics*.

Lawton, J., Beard, R., and Young, B. (2003). A decentralized approach to formation maneuvers. *IEEE Transactions on Robotics and Automation*, 19(6), 933–941.

Paliotta, C., Lefeber, E., Pettersen, K.Y., Pinto, J., Costa, M., and de Figueiredo Borges de Sousa, J.T. (2019). Trajectory tracking and path following for underactuated marine vehicles. *IEEE Transactions on Control Systems Technology*.

Saska, M., Spurný, V., and Vonásek, V. (2016). Predictive control and stabilization of nonholonomic formations with integrated spline-path planning. *Robotics and Autonomous Systems*, 75, 379–397.

Sousa, A., Madureira, L., Coelho, J., Pinto, J., Pereira, J., Borges Sousa, J., and Dias, P. (2012). LAUV: The manportable autonomous underwater vehicle. In *Proc. 3rd IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles*.

Van Parys, R. and Pipeleers, G. (2017). Distributed MPC for multi-vehicle systems moving in formation. *Robotics and Autonomous Systems*, 97, 144–152.

Wang, Y., Yang, Y., Pu, Y., and Manzie, C. (2021). Path following by formations of agents with collision avoidance guarantees using distributed model predictive control. In *Proc. 2021 American Control Conference*.

Zhu, Z., Hu, S.L.J., and Li, H. (2016). Effect on Kalman based underwater tracking due to ocean current uncertainty. In *Proc. 2016 IEEE/OES Autonomous Underwater Vehicles*, 131–137.