2018 European Control Conference (ECC)
June 12-15, 2018. Limassol, Cyprus

Distributed model predictive control of multiple vehicles transporting a flexible payload*

Sampath Kumar Mulagaleti[2], Ruben Van Parys[1] and Goele Pipeleers[1]

Abstract— **This paper presents a control strategy for multiple vehicles that cooperatively transport a flexible payload. To this end, an algorithm is developed which generates optimal trajectories for the vehicles to follow. Solving an optimization problem composes the core of the algorithm. The problem is first decomposed over the vehicles using the Alternating Direction Method of Multipliers (ADMM) algorithm. This results in each vehicle solving a sub-problem to generate its own optimal trajectory. The algorithm instructs that the optimization problem be solved repeatedly in a receding horizon fashion, making it fit into a distributed model predictive control (DMPC) framework. One ADMM iteration is performed per DMPC iteration, reducing the inter-agent communication rate. Numerical validation of the developed control scheme is performed and the results are presented.**

I. INTRODUCTION

Payload transportation tasks are ubiquitous in industrial environments. These payloads can vary greatly in weights, necessitating multiple transportation solutions of different payload capacities within the same logistics structure. An alternative to this would be to use multiple vehicles to cooperatively tow a payload, with the number of vehicles chosen according to the towing requirements. To enable such a multi-agent system to work autonomously and transport the payload to a desired location, control strategies are required which perform coordinated motion of the vehicles.

Transportation of objects with multiple vehicles is a standard problem in robotics and control. Some of the early approaches to model and control such systems are presented in [1], [2], [3], and an optimal control approach in [4]. More recent approaches are formulated in a Model Predictive Control (MPC) framework, which allows to explicitly incorporate system constraints in the optimization problem. Such techniques have been applied to perform trajectory tracking with multiple vehicles moving in a desired formation [5]. Motion planning for multiple vehicles moving in a formation is done within a receding horizon framework in [6], utilizing a leader-follower strategy. Such a strategy is not robust against possible failure of the leader agent. This necessitates development of strategies in which each agent performs an equivalent task within the multi-agent system. This leads to the domain of Distributed Model Predictive Control (DMPC). Within the DMPC framework, the control problem is distributed over the agents. Since an optimization problem underlies the MPC framework, DMPC based techniques usually employ distributed optimization methods to distribute the optimization problem over the agents [7]. Motion planning algorithms are designed within this framework in [8] and [9] for formation control of multiple agents.

Unlike formation control problems, agents in payload transportation problems are dynamically coupled. This means that a change in state of one of the agents alters the state of the others. The coupled dynamics act as constraints shared by all the agents within the multi-agent system, and decomposing the centralized problem for DMPC purposes would require decomposing the coupling dynamics constraint. Decoupling of dynamics is done using primal decomposition in [10], in which, linear systems are considered, and an optimal consensus problem is solved. In optimal consensus problems, multiple optimization variables are driven to the same value over iterations. Another approach based on dual decomposition, is discussed in [11]. In this work, the variable splitting method of [8] is employed. According to this method, copies of states of a multi-agent system that are not local to a constituent vehicle are introduced as exogenous inputs on the vehicle. More robust methods based on the ADMM are formulated in [12], which consider coupled objective functions in addition to coupled nonlinear dynamics. Another alternative is introduced in [13], where the ADMM-consensus problem introduced in [14] is combined with the idea of introducing copies of neighboring states from [11].

This work extends the DMPC scheme introduced for the formation control problem in [9], to the cooperative payload transportation problem. The main advantage of this scheme is that it achieves a decentralized-consensus in a dynamically coupled system, by decoupling the 2nd ADMM step using a novel variable-copying scheme. In order to reduce the communication and computation load, 1 ADMM iteration is performed per DMPC iteration, and the ADMM iterations are supported by inter-vehicle communication. The proposed scheme is implemented using a spline-based parameterization discussed in [9], and a numerical validation of the algorithm is performed.

Section II introduces the motion planning problem for cooperative payload transportation problem, and Section III describes how this problem is solved in an online distributed manner. Results from simulations based on the proposed

*This work benefits from KU Leuven-BOF PFV/10/002 Centre of Excellence: Optimization in Engineering (OPTEC); from the project G0C4515N of the Research Foundation-Flanders (FWO-Flanders); and from the KU Leuven Research project C14/15/067: B-spline based certificates of positivity with applications in engineering. Ruben Van Parys is a PhD fellow of FWO-Flanders. Flanders Make is the Flemish strategic research centre for the manufacturing industry.

[1]The authors are with the MECO Research Team, Department Mechanical Engineering, KU Leuven and with the DMMS lab, Flanders Make, Leuven, Belgium. [2]The author is with the IMT School for Advanced Studies Lucca, 55100, Italy. Email: s.mulagaleti@imtlucca.it

978-3-9524-2699-9 ©2018 EUCA

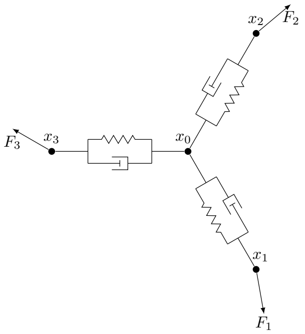

Fig. 1: Dynamic model decoupling

approach are presented in Section IV, and the conclusions in Section V.

II. PROBLEM FORMULATION

A. System model

The dynamic system under consideration here is shown in Figure 1. It is modeled as multiple holonomic vehicles attached through 1D spring-damper elements to a common payload. The payload and the holonomic vehicles are abstracted as point masses, and are constrained to move in a 2D plane. Dynamic spring-mass-damper models are used to describe the system in the optimization problem. Coulomb friction is ignored in the model. The free length of the flexible elements is assumed to be 0 m, resulting in the vehicles only applying a pulling force on the payload. In this paper, N=3 vehicles are considered.

B. Optimal control problem

This section discusses the formulation of the considered optimal control problem. The optimization variables are the payload and the vehicle trajectories. The solution of this problem are the trajectories that transport the payload to a location closest to its goal $x_d \in \mathbb{R}^2$ within the time horizon T over which the problem is solved. The payload trajectory is represented by the vector $x_0 : [0,T] \to \mathbb{R}^2$ and that of each vehicle i by $x_i : [0,T] \to \mathbb{R}^2$. The formulation is shown in (1).

$$\begin{aligned} \underset{x_0(.),x_1(.),x_2(.),x_3(.)}{\text{minimize}} \quad & \int_0^T ||x_0(t)-x_d||_1 dt \\ \text{subject to} \quad & g(x_0(t),x_1(t),x_2(t),x_3(t)) = 0 \\ & h_{ij}(x_0(t),x_i(t),x_j(t)) \leq 0 \\ & \underline{l} \preceq f_i(x_0(t),x_i(t)) \preceq \bar{l} \\ & \underline{d} \preceq d_i(x_i(t)) \preceq \bar{d} \\ & \{x_0^{(k)}(0), x_0^{(k)}(T)\} = X^p \\ & \{x_i^{(k)}(0), x_i^{(k)}(T)\} = X^i \\ & \forall t \in [0,T], \quad \forall i,j \in \{1,2,3\}, i \neq j \end{aligned} \tag{1}$$

As mentioned earlier, the coupled payload dynamics of the multi-agent system are modeled as constraints in the optimization problem. It is expressed in (2). The time dependence of the trajectories is dropped for ease of notation. The parameters m_i and m_0 represent vehicle i's and payload's mass respectively, and c_i and k_i represent damping and stiffness coefficients of flexible element i.

$$g(x_0,x_1,x_2,x_3) = m_0\ddot{x}_0 + \sum_{i=1}^{3} \left[c_i(\dot{x}_i - \dot{x}_0) + k_i(x_i - x_0)\right] \tag{2}$$

Intra-vehicle collisions are avoided by the function h_{ij}, expressed in (3), utilizing the inner-product notation. It limits the angles between the vehicles to lie at greater than 90° when subtended at the payload. Unlike the complete coupled constraint g, this is a constraint involving nearest neighbors.

$$h_{ij}(x_0,x_i,x_j) = \langle x_i - x_0, x_j - x_0 \rangle \tag{3}$$

The function f_i consists of individual vehicle i's dynamics, and flexible element i's length. The lower and upper bounds on these quantities are captured in vectors $\underline{l}$ and $\bar{l}$ respectively. This function is shown in (4).

$$f_i(x_0,x_i) = \begin{bmatrix} m_i\ddot{x}_i + c_i(\dot{x}_i - \dot{x}_0) + k_i(x_i - x_0) \\ \langle x_i - x_0, x_i - x_0 \rangle \end{bmatrix} \tag{4}$$

Limitations on vehicle velocities and accelerations are expressed through the function d_i shown in (5), with the limits on these quantities captured in the vectors $\underline{d}$ and $\bar{d}$.

$$d_i(x_i) = \begin{bmatrix} \dot{x_i}^T & \ddot{x_i}^T \end{bmatrix}^T \tag{5}$$

The remaining constraints represent limits on vehicle velocities and accelerations, and initial and final conditions on the vehicle and payload position trajectories and higher derivatives.

III. SOLUTION STRATEGY

A. Spline parameterization

Problem (1) represents an infinite dimensional problem, since the optimization variables $x_0(.)$ and $x_i(.)$ are infinite dimensional, and enforcing the constraints at all time results in an infinite number of constraints. In order to make the problem numerically tractable, a B-spline based parameterization of the optimization variables is used, along the lines of [15]. The continuous optimization variables are hence represented as linear combinations of piecewise polynomial B-spline basis functions $b_k(t)$, as expressed in (6).

$$x_i(t) = \sum_{k=1}^{n} \mathsf{x}_{i,k} b_k(t) = \mathsf{x}_i^T b(t) \quad \forall i \in \{0,1,.,N\} \tag{6}$$

The trajectories of each vehicle i are consequently represented by the coefficient set x_i= $\{\mathsf{x}_{i,k}\}_{k=1}^n$ and those of the payload by x_0=$\{\mathsf{x}_{0,k}\}_{k=1}^n$, for a chosen B-spline basis functions vector $b = [b_1, \ldots, b_n]^T$. Substituting these parameterizations in (1) results in an optimization problem with finite number of variables. A B-spline parameterization is chosen because of its convex hull property [15], which dictates that the resultant spline function lies within the convex hull of the

Algorithm 1 Spline-based MPC

```
Repeat every $\Delta T$: $k = 0, 1, \ldots$
    Extract $x_i^k(t)$ from $\mathsf{x}_i^k$ for each Vehicle $i$.
    Vehicle $i$ starts following trajectory $x_i^k(t)$
    Estimate $\hat{x}_i^k$ and $\hat{x}_0^k$ at time $(k+1)\Delta T$
    Compute $\tilde{\mathsf{x}}_i^k$ and $\tilde{\mathsf{x}}_0^k$ over the updated horizon.
    Compute $\mathsf{x}_i^{k+1}$ and $\mathsf{x}_0^{k+1}$ by solving (7), using $\hat{x}_i^k$,
    $\hat{x}_0^k$ as initial conditions and $\tilde{\mathsf{x}}_i^k$ as hot start coefficients
Until target reached
```

coefficients. Hence, the constraints on an infinite dimensional spline can be imposed on its coefficients, resulting in a finite number of constraints. Reducing potential conservatism is discussed in [16]. Since the constraint functions f_i, g and h_{ij} are composed of polynomial combinations of the splines $x_i(t)$ and $x_0(t)$ and their derivatives, the resultant functions are splines in a B-spline polynomial basis as well. The coefficients of these splines are polynomial combinations of x_0 and x_i. Following the convex hull property, these constraints are transformed to f_i, g and h_{ij} respectively. The coefficients, and functions composed of those coefficients are indicated in a sans-serif font. Non-coupled constraints on the spline coefficients are captured in X_i and X_0. The resulting formulation is seen in (7).

$$\begin{aligned} \underset{\forall i: \mathsf{x}_i, \mathsf{x}_0}{\text{minimize}} \quad & \mathsf{J}(\mathsf{x}_0) \\ \text{subject to} \quad & \mathsf{g}(\mathsf{x}_0, \mathsf{x}_1, \mathsf{x}_2, \mathsf{x}_3) = 0 \\ & \mathsf{h}_{ij}(\mathsf{x}_0, \mathsf{x}_i, \mathsf{x}_j) \leq 0 \\ & \underline{l} \preceq \mathsf{f}_i(\mathsf{x}_0, \mathsf{x}_i) \preceq \bar{l} \\ & \mathsf{x}_i \in \mathsf{X}_i \\ & \mathsf{x}_0 \in \mathsf{X}_0 \\ & \forall i, j \in \{1, 2, 3\}, i \neq j \end{aligned} \tag{7}$$

The MPC updating scheme proposed in [9] for optimization variables expressed in a B-spline basis can be used on a centralized processor, provided it has access to all x_i and x_0 variables. This scheme is summarized in Algorithm 1. The vehicle and payload positions that are estimated at time $(k+1)\Delta T$ are incorporated into the feasible sets X_i and X_0 while solving the problem for x_i^{k+1} and x_0^{k+1}. The hot-start coefficients are $\tilde{\mathsf{x}}_i^k$ and $\tilde{\mathsf{x}}_0^k$ are obtained by expressing x_i^k and x_0^k in current MPC horizon's basis, as discussed in [9].

B. Distributed formulation

The purpose of this section is to split the optimization problem (7) into 3 separate problems, one for each vehicle. This allows each vehicle to solve for its own trajectory, aided by communication between the vehicles. The first step is the introduction of copies of the payload variables x_0 on each vehicle, represented by x_{i0}. This substitution results in 3 copies of the payload dynamics constraint g, each one being $\mathsf{g}(\mathsf{x}_{i0}, \mathsf{x}_i, \mathsf{x}_j) = 0$, where $\mathsf{x}_j = \{\mathsf{x}_1, \mathsf{x}_2, \mathsf{x}_3\} \ominus \mathsf{x}_i$. The symbol $\ominus$ denotes the set-difference operator.

The payload dynamics constraint is further decoupled by borrowing ideas from [11], which dictates that the variables $x_j(t)$ of vehicle j which affect the dynamics of vehicle i can be modeled as exogenous inputs $x_{ij}(t)$ acting on vehicle i. The new variables $x_{ij}(t)$ are hence local to vehicle i, and are parameterized by the spline coefficients x_{ij}. The payload dynamics constraint is thus further modified as $\mathsf{g}(\mathsf{x}_{i0}, \mathsf{x}_i, \mathsf{x}_{ij}) = 0$. This is supported by consensus constraints between x_j and x_{ij}. Similar substitutions are performed in the anti-collision constraints h_{ij}. Following a modification of the consensus constraints presented in [9], consensus variables z_{ij} are introduced on each vehicle i, which act as a balance between x_j and x_{ij}. The modified optimization problem with local copy variables, consensus variables, and consensus constraints is shown in (8).

$$\begin{aligned} \underset{\forall i,j,i\neq j: \mathsf{x}_i, \mathsf{x}_{ij}, \mathsf{z}_{ij}, \mathsf{x}_{i0}}{\text{minimize}} \quad & \sum_{i=1}^{3} \mathsf{J}(\mathsf{x}_{i0}) \\ \text{subject to} \quad & \underline{l} \preceq \mathsf{f}_i(\mathsf{x}_{i0}, \mathsf{x}_i) \preceq \bar{l} \\ & \mathsf{g}(\mathsf{x}_{i0}, \mathsf{x}_i, \mathsf{x}_{ij}) = 0 \\ & \mathsf{h}_{ij}(\mathsf{x}_{i0}, \mathsf{x}_i, \mathsf{x}_{ij}) \leq 0 \\ & \mathsf{x}_{ij} = \mathsf{z}_{ij},\ \mathsf{x}_j = \mathsf{z}_{ij} \\ & \mathsf{x}_i \in \mathsf{X}_i,\ \mathsf{x}_{i0} \in \mathsf{X}_0 \\ & \forall i, j \in \{1, 2, 3\}, i \neq j \end{aligned} \tag{8}$$

The optimization problem (8) can now be decoupled using the Alternating Direction Method of Multipliers (ADMM). The ADMM algorithm solves the dual of (8), after augumenting it with a quadratic term to improve convergence properties. To minimize the duality gap, the fewest number of constraints are dualized according to [8]. Hence, only the consensus equality constraints are moved into the objective function to form the Lagrangian:

$$\begin{aligned} \mathcal{L}_\rho = & \sum_{i=1}^{3} \Big(\mathsf{J}_i(\mathsf{x}_{i0}) + \sum_{j\neq i} \big(\lambda_{ij}^T (\mathsf{x}_{ij} - \mathsf{z}_{ij}) + \frac{\rho}{2} ||\mathsf{x}_{ij} - \mathsf{z}_{ij}||_2^2 \big) \\ & \qquad + \sum_{j\neq i} \big(\mu_{ij}^T (\mathsf{x}_j - \mathsf{z}_{ij}) + \frac{\rho}{2} ||\mathsf{x}_j - \mathsf{z}_{ij}||_2^2 \big) \Big) \\ = & \sum_{i=1}^{3} \mathcal{L}_{\rho,i}(\mathsf{x}_{i0}, \mathsf{x}_{ij}, \mathsf{z}_{ij}, \lambda_{ij}, \mathsf{x}_j, \mathsf{z}_{ij}, \mu_{ij}) \\ = & \sum_{i=1}^{3} \mathcal{L}_{\rho,i}(\mathsf{x}_{i0}, \mathsf{x}_{ij}, \mathsf{z}_{ij}, \lambda_{ij}, \mathsf{x}_i, \mathsf{z}_{ji}, \mu_{ji}) \end{aligned} \tag{9}$$

The dual variables introduced in the Lagrangian formulation corresponding to the equality constraints are λ_{ij} and μ_{ij}. The last equality in (9) is valid due to the bidirectionality of interaction amongst the vehicles: variables on vehicle i affecting the dynamics of vehicle j correspond to the variables on vehicle j affecting dynamics of vehicle i. Since the constraints are now completely decoupled, the feasible set on each vehicle i is represented by $\{\mathsf{x}_i, \mathsf{x}_{i0}, \mathsf{x}_{ij}\} \in \Phi_i$.

The dual function of (8) is shown in (10).

$$q(\lambda_{ij},\mu_{ij}) = \inf_{\substack{\{\mathsf{x}_i,\mathsf{x}_{i0},\mathsf{x}_{ij}\}\in\Phi_i \\ \forall i,j\in\{1,2,3\},i\neq j}} \mathcal{L}_\rho \tag{10}$$

The dual variables are found by maximizing the dual function q with respect to the lagrangian multipliers. The maximization is performed through gradient ascent in the ADMM algorithm. For this, the primal variables x^k_{ij},x^k_j and z^k_{ij} from iteration k are updated in iteration $k+1$, and the gradient of the dual function q is calculated in the directions of the lagrangian multipliers according to (11).

$$\begin{aligned}\nabla_{\lambda_{ij}} q &= \mathsf{x}^{k+1}_{ij} - \mathsf{z}^{k+1}_{ij} \\ \nabla_{\mu_{ij}} q &= \mathsf{x}^{k+1}_{j} - \mathsf{z}^{k+1}_{ij}\end{aligned} \tag{11}$$

An optimal step of length ρ in these directions is taken. The primal variables are updated in two steps in a Gauss-Seidel fashion, the first step updating the $\{\mathsf{x}_i,\mathsf{x}_{i0},\mathsf{x}_{ij}\}$ variables and the second updating the z_{ij} variables.

Solving the optimization problem (7) till convergence will require several ADMM iterations. These iterations will replace (step 5) in Algorithm 1, and would be performed on individual vehicles in parallel. However, this would also result in a substantial computation and communication load for each MPC iteration. This is avoided by following the DMPC scheme proposed in [9], which performs 1 ADMM iteration per MPC iteration. The resultant is Algorithm 2, which includes the communication steps performed in one DMPC iteration. The algorithm dictates that first a few ADMM iterations be performed to find the first set of primal and dual variables. The primal x_i variables are then interpreted in terms of the trajectories $x_i(t)$ on each vehicle i, and the tracking of these trajectories begins. According to the spline-MPC scheme discussed in [9], $\hat{x}_i(\Delta T)$ and $\hat{x}_{i0}(\Delta T)$ are estimated locally on each vehicle, with perfectly tracked $x_{ij}(t)$ as the exogenous inputs into the local model. These estimates are passed as initial conditions into the MPC problem solving the trajectories from time ΔT. This is followed by the usual 3 ADMM steps, with steps 7 and 9 for calculating the primal variables, and step 10 for the dual variables. Since variables from the previous ADMM iteration are used in the current iteration, the spline coefficients are transformed to represent the spline in the current time frame, as discussed in [9]. Step 6 of the algorithm deals with this transformation. Step 9 is the direct least squares solution of the second ADMM step to update the z_{ij} variables, and the corresponding problem is shown in (12).

$$\begin{aligned}\mathsf{z}^{k+1}_{ij} := \underset{\mathsf{z}_{ij}}{\operatorname{argmin}} \quad & (\tilde{\lambda}^k_{ij})^T(\mathsf{x}^{k+1}_{ij} - \mathsf{z}_{ij}) + \frac{\rho}{2}||\mathsf{x}^{k+1}_{ij} - \mathsf{z}_{ij}||^2_2 \\ & + (\tilde{\mu}^k_{ij})^T(\mathsf{x}^{k+1}_{j} - \mathsf{z}_{ij}) + \frac{\rho}{2}||\mathsf{x}^{k+1}_{j} - \mathsf{z}_{ij}||^2_2\end{aligned} \tag{12}$$

From this least squares problem, it can be seen the consensus variables z_{ij} are pushed between the *expected* x_{ij} variables and the *actual* x_j variables, acting as a balance between them. In the first ADMM step (Step 7), the reverse occurs in that x_{ij} and x_j are pushed towards the previous *compromise* z_{ij} variables. Thus, over MPC (and hence, ADMM) iterations, a consensus is achieved between x_{ij} and x_j, resulting the problem dynamics being satisfied.

Algorithm 2 ADMM based Distributed MPC

```
1:  Perform $n$ ADMM iterations and get $\mathsf{x}^0_i$, $\mathsf{x}^0_{ij}$, $\mathsf{x}^0_{i0}$, $\mathsf{z}^0_{ij}$, $\lambda^0_{ij}$, $\mu^0_{ij}$
2:  **Repeat** every $\Delta T$: $k = 0, 1, \ldots$
3:      Extract $x^k_i(t)$, $x^k_{ij}(t)$ and $x^k_{i0}(t)$ from $\mathsf{x}^k_i$, $\mathsf{x}^k_{ij}$ and $\mathsf{x}^k_{i0}$ on each Vehicle $i$.
4:      Vehicle $i$ starts following trajectory $x^k_i(t)$
5:      Estimate $\hat{x}^k_i$ and $\hat{x}^k_{i0}$ at time $(k+1)\Delta T$, assuming perfect tracking of $x^k_{ij}(t)$
6:      Update horizon and compute $\tilde{\mathsf{x}}^k_i$, $\tilde{\mathsf{x}}^k_{ij}$, $\tilde{\mathsf{x}}^k_{i0}$, $\tilde{\mathsf{z}}^k_{ij}$, $\tilde{\mathsf{z}}^k_{ji}$, $\tilde{\lambda}^k_{ij}$, $\tilde{\mu}^k_{ji}$
7:      Compute $\{\mathsf{x}^{k+1}_i, \mathsf{x}^{k+1}_{ij}, \mathsf{x}^{k+1}_{i0}\}$, using $\hat{x}^k_i$ and $\hat{x}^k_{i0}$ as initial conditions:
        $\begin{pmatrix}\mathsf{x}^{k+1}_i \\ \mathsf{x}^{k+1}_{ij} \\ \mathsf{x}^{k+1}_{i0}\end{pmatrix} := \underset{\{\mathsf{x}_i,\mathsf{x}_{i0},\mathsf{x}_{ij}\}\in\Phi_i}{\operatorname{argmin}} \mathcal{L}_{\rho,i}(\mathsf{x}_{i0}, \mathsf{x}_{ij}, \tilde{\mathsf{z}}^k_{ij}, \tilde{\lambda}^k_{ij}, \mathsf{x}_i, \tilde{\mathsf{z}}^k_{ji}, \tilde{\mu}^k_{ji})$
8:      Communication with agents j: :
            send $\mathsf{x}^{k+1}_i$, receive $\mathsf{x}^{k+1}_j$
9:      Compute $\mathsf{z}^{k+1}_{ij}$:
        $\mathsf{z}^{k+1}_{ij} := \frac{1}{2}\left(\mathsf{x}^{k+1}_j + \mathsf{x}^{k+1}_{ij} + \frac{\tilde{\lambda}^k_{ij} + \tilde{\mu}^k_{ij}}{\rho}\right)$
10:     Compute $\lambda^{k+1}_{ij}$ and $\mu^{k+1}_{ij}$:
        $\lambda^{k+1}_{ij} := \tilde{\lambda}^k_{ij} + \rho(\mathsf{x}^{k+1}_{ij} - \mathsf{z}^{k+1}_{ij})$
        $\mu^{k+1}_{ij} := \tilde{\mu}^k_{ij} + \rho(\mathsf{x}^{k+1}_{j} - \mathsf{z}^{k+1}_{ij})$
11:     Communication with agents $j$
            send $\mathsf{z}^{k+1}_{ij}$ and $\mu^{k+1}_{ij}$, receive $\mathsf{z}^{k+1}_{ji}$ and $\mu^{k+1}_{ji}$
12: **Until** target reached
```

IV. Simulation results

The DMPC strategy proposed in Algorithm 2 is numerically validated for the multi-agent system with parameters m_0= 5kg, m_i=2kg, c_i=20Ns/m and K_i=50N/m. The velocities and accelerations of the vehicles are enforced to start from $0m/s$ and $0m/s^2$ respectively. The initial positions have been chosen such that the payload starts from a balanced state. The vehicle and payload trajectories are parameterized as B-splines with cubic basis functions having 10 knot intervals. This parameterization avoids conservatism. The time horizon is T = 5s, and each DMPC update happens every ΔT = 0.1s. The ADMM penalty parameter is ρ = 0.5. The optimization problem in step 7 of Algorithm 2 is solved using the interior point solver IPOPT [17], for which the exact Jacobians and Hessians are provided by CasADi [18]. The initial conditions for each DMPC update in the simulations are obtained following the assumption of perfect tracking of trajectories calculated in the previous DMPC update by the vehicles.

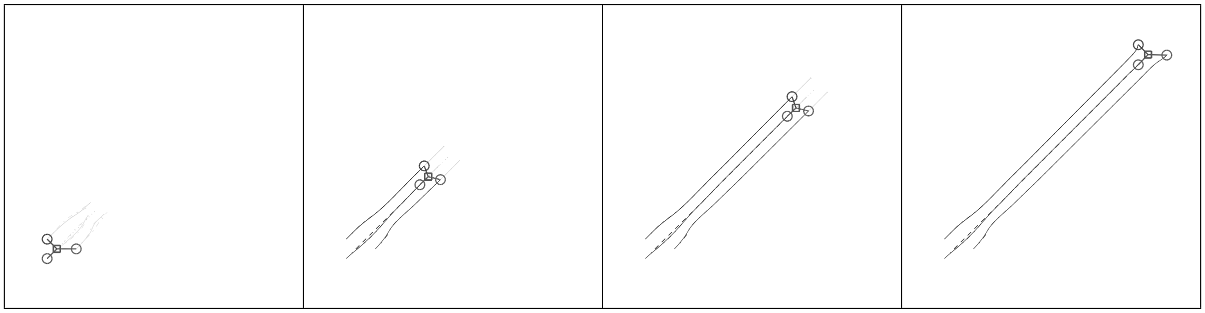

Fig. 2: Motion trajectories for the payload and holonomic vehicles within the multi-agent system. The dotted lines represent payload trajectory. The gray lines indicate the computed trajectories to follow. The black lines represent the covered paths.

A. Payload transportation in an obstacle-free environment

The motion trajectories of the vehicles and payload upon implementation of Algorithm 2 to control the multi-agent system are shown in Figure 2. The darker lines represent trajectories that have been traversed by the vehicles and payload, and the lighter lines represent future trajectories. The corresponding velocity trajectories tracked by the vehicles are shown in Figure 3.

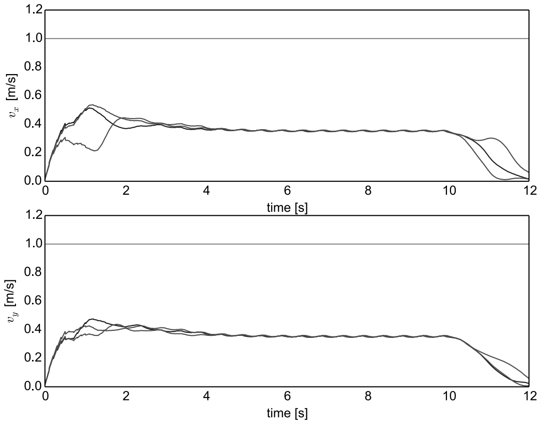

Fig. 3: Velocity trajectories of the vehicles. Dark line at 1 m/s indicates constraint.

Since consensus constraints such as $\mathsf{x}_{i0} = \mathsf{x}_{j0}$ are not imposed, the payload positions are free to evolve according to the local dynamics of each vehicle. However, the complete multi-agent system dynamics are satisfied only if the local payload position copies converge towards each other. Hence, convergence of local payload positions towards each other over successive ADMM iterations is an indicator for dynamics satisfaction. This convergence is monitored by using distance metrics between the payload trajectory coefficients x_{10}, x_{20} and x_{30}, calculating a measure of dynamics error ϵ_0 as:

$$\epsilon_0 = \sum_{i=1}^{3} \sum_{j \neq i} \langle \mathsf{x}_{i0} - \mathsf{x}_{j0}, \mathsf{x}_{i0} - \mathsf{x}_{j0} \rangle \tag{13}$$

For the trajectories shown in Figure 2, the corresponding ϵ_0 is shown in Figure 4.

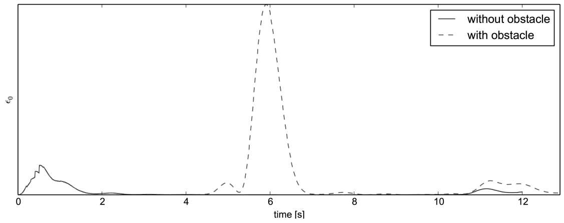

Fig. 4: Error in system dynamics.

Since all vehicles start with the same initial value of the payload position, the coefficients match resulting in ϵ_0=0. However, the local payload copies diverge from each other during the initial few DMPC iterations, and before halt of the payload. This is because each vehicle i is agnostic of each other vehicle j's force limits. At the start, the vehicle i requests a trajectory x_{ij} of vehicle j which results in an immediate pull of the payload with a high force. Towards the end, the opposite occurs with each vehicle i requesting a x_{ij} pulling the payload to an immediate halt. However, vehicle j can realistically only provide a trajectory x_j while respecting its own constraints. This trajectory is different from the requested x_{ij}, resulting in a loss of consensus. The local payload dynamics thus evolve with a different set of inputs on each vehicle, increasing the value of ϵ_0. This disagreement is however penalized through the consensus variables z_{ij}, and dual variables λ_{ij} and μ_{ij}. The penalization results in the vehicles achieving consensus on the payload position over successive DMPC iterations, reducing the value of ϵ_0. Hence, it is concluded that the dynamics of the complete multi-agent system are satisfied. The number of iterations after which consensus is achieved can be reduced by increasing the value of the ADMM penalty parameter ρ. However, this results in lower vehicle velocities indicating a loss in optimality with respect to payload transportation.

The ability to achieve consensus can be seen as an indication of convergence of the real-time ADMM scheme. This is as expected, since the current optimization problem is convex. Non-holonomic vehicle models might introduce nonconvexity into the problem, resulting in a loss of convergence. Further investigation into these properties is currently being performed, in addition to robustness with respect to noises, parameter variabilities and communication delays.

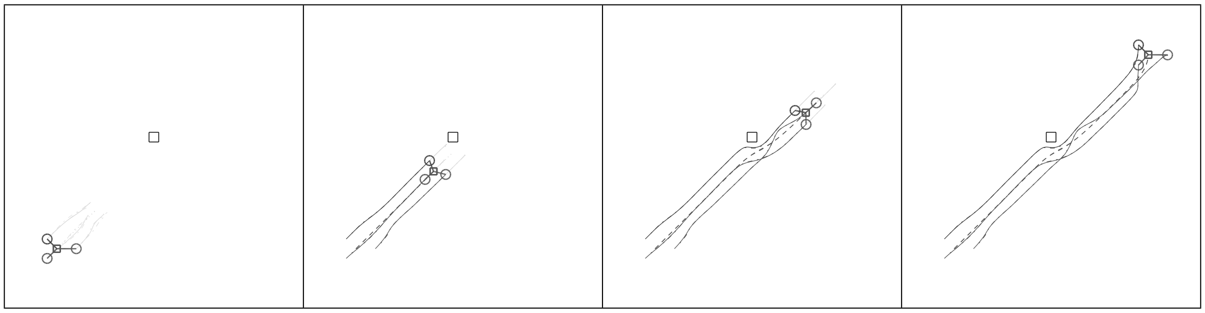

Fig. 5: Motion trajectories of the vehicles and payload in the presence of a stationary obstacle.

B. Payload transportation with obstacle avoidance

Static obstacle avoidance behavior is obtained by adding additional spline variables representing separating hyperplane coefficients, as discussed in [15]. Each vehicle i is assigned a set of these variables $a_i(t)$ and $b_i(t)$, for which the separating hyperplane constraints are defined. These constraints make sure that the trajectories $x_i(t)$, $x_{i0}(t)$ and $x_{ij}(t)$ lie on one side of the separating hyperplane parameterized by $a_i(t)$ and $b_i(t)$, and the obstacle lies on the other side of the hyperplane. The trajectories generated with this behavior is shown in Figure 5. Near the obstacle, there is a loss of consensus between x_{ij} and x_j, which is regained over successive ADMM iterations. Like before, the rate at which the consensus is regained can be controlled by tuning the ADMM penalty parameter ρ.

V. Conclusion and future work

This paper presents an algorithm to solve the multi-vehicle payload transportation problem in a Distributed Model Predictive Control (DMPC) framework. The optimization problem underlying the control problem is decoupled over the multiple vehicles based on the Alternating Direction Method of Multipliers (ADMM). A novel variable copying scheme is adopted, which results in a decentralized consensus formulation for a dynamically coupled system. Results based on numerical simulations are observed, and consensus on payload position is noted leading to the conclusion that the coupled dynamics are satisfied over successive iterations of the DMPC algorithm. It is noted that algorithms developed along the same lines can be applied to systems with $N \geq 2$ vehicles. Future work includes modifying the algorithm to include dynamics considering free length of the flexible elements, which makes practical validation with real-life holonomic vehicles feasible. Further modifications can be done to the algorithm to analyze the effects of additional constraints imposing consensus on local payload copies and separating hyperplane variables on dynamic behavior.

References

[1] D. Williams and O. Khatib, "The virtual linkage: a model for internal forces in multi-grasp manipulation," in *[1993] Proceedings IEEE International Conference on Robotics and Automation*, pp. 1025–1030 vol.1, May 1993.

[2] O. Khatib, K. Yokoi, K. Chang, D. Ruspini, R. Holmberg, A. Casal, and A. Baader, *Force Strategies for Cooperative Tasks in Multiple Mobile Manipulation Systems*, pp. 333–342. London: Springer London, 1996.

[3] H. G. Tanner, K. J. Kyriakopoulos, and N. J. Krikelis, "Modeling of multiple mobile manipulators handling a common deformable object," *Journal of Robotic Systems*, vol. 15, no. 11, pp. 599–623, 1998.

[4] R. Ritz and R. D'Andrea, "Carrying a flexible payload with multiple flying vehicles," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3465–3471, Nov 2013.

[5] S. Liu, D. Sun, and C. Zhu, "Coordinated motion planning for multiple mobile robots along designed paths with formation requirement," *IEEE/ASME Transactions on Mechatronics*, vol. 16, pp. 1021–1031, Dec 2011.

[6] M. Saska, V. Vonásek, T. Krajník, and L. Přeučil, "Coordination and navigation of heterogeneous mav-ugv formations localized by a 'hawk-eye'-like approach under a model predictive control scheme," *Int. J. Rob. Res.*, vol. 33, pp. 1393–1412, Sept. 2014.

[7] S. Boyd, L. Xiao, and A. Mutapcic, "Notes on decomposition methods," October 2003.

[8] R. L. Raffard, C. J. Tomlin, and S. P. Boyd, "Distributed optimization for cooperative agents: application to formation flight," in *2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601)*, vol. 3, pp. 2453–2459 Vol.3, Dec 2004.

[9] R. Van Parys and G. Pipeleers, "Distributed MPC for multi-vehicle systems moving in formation," *Robotics and Autonomous Systems*, vol. 97, no. Supplement C, pp. 144 – 152, 2017.

[10] B. Johansson, A. Speranzon, M. Johansson, and K. H. Johansson, "On decentralized negotiation of optimal consensus," *Automatica*, vol. 44, pp. 1175–1179, Apr. 2008.

[11] P. Giselsson and A. Rantzer, "Distributed model predictive control with suboptimality and stability guarantees," in *49th IEEE Conference on Decision and Control (CDC)*, pp. 7272–7277, Dec 2010.

[12] F. Farokhi, I. Shames, and K. H. Johansson, *Distributed MPC Via Dual Decomposition and Alternative Direction Method of Multipliers*, pp. 115–131. Dordrecht: Springer Netherlands, 2014.

[13] T. H. Summers and J. Lygeros, "Distributed model predictive consensus via the alternating direction method of multipliers," in *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 79–84, Oct 2012.

[14] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, pp. 1–122, Jan. 2011.

[15] T. Mercy, R. Van Parys, and G. Pipeleers, "Spline-based motion planning for autonomous guided vehicles in a dynamic environment," *IEEE Transactions on Control Systems Technology*, vol. PP, no. 99, pp. 1–8, 2017.

[16] W. Van Loock, G. Pipeleers, and J. Swevers, "B-spline parameterized optimal motion trajectories for robotic systems with guaranteed constraint satisfaction," *Mechanical Sciences*, vol. 6, no. 2, pp. 163–171, 2015.

[17] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, no. 1, pp. 25–57, 2006.

[18] J. Andersson, *A general-purpose software framework for dynamic optimization*. PhD thesis, Arenberg Doctoral School, KU Leuven, Department of Electrical Engineering (ESAT/SCD) and Optimization in Engineering Center, Kasteelpark Arenberg 10, 3001-Heverlee, Belgium, 2013.