2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
September 27 - October 1, 2021. Prague, Czech Republic

2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) | 978-1-6654-1714-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/IROS51168.2021.9636444

Learning Environment Constraints in Collaborative Robotics: A Decentralized Leader-Follower Approach

Monimoy Bujarbaruah[1], Yvonne R. Stürz[1], Conrad Holda[1], Karl H. Johansson[2], Francesco Borrelli[1]

***Abstract*— In this paper, we propose a leader-follower hierarchical strategy for two robots collaboratively transporting an object in a partially known environment with obstacles. Both robots sense the local surrounding environment and react to obstacles in their proximity. We consider no explicit communication, so the local environment information and the control actions are not shared between the robots. At any given time step, the leader solves a model predictive control (MPC) problem with its known set of obstacles and plans a feasible trajectory to complete the task. The follower estimates the inputs of the leader and uses a policy to assist the leader while reacting to obstacles in its proximity. The leader infers obstacles in the follower's vicinity by using the difference between the predicted and the real-time estimated follower control action. A method to switch the leader-follower roles is used to improve the control performance in tight environments. The efficacy of our approach is demonstrated with detailed comparisons to two alternative strategies, where it achieves the highest success rate, while completing the task fastest.**

I. Introduction

Multi-robot systems have a high potential to collaboratively accomplish complex tasks, such as, for example transporting large or heavy work pieces [1]–[7]. For known repetitive tasks in structured environments, collaborative manipulation problems in industrial applications are mainly solved in a centralized way, relying on precise feedforward computations. As solving a centralized control synthesis problem for such tasks can become computationally cumbersome, distributed and decentralized control strategies have been proposed [8]–[14]. However, communication delays remain as a bottleneck, i.e., iterative communication, such as the ones required for consensus type algorithms, might converge too slowly for such robotics applications. To resolve this, the use of implicit communication, such as force and torque measurements or estimates on rigid bodies, has been proposed in the literature, e.g., in [15]–[24].

Obstacle avoidance in collaborative robotics has primarily considered known obstacles and solving a centralized problem with explicit communication [25]–[28]. Issues arise when the robots rely only on local controllers in unstructured and unknown or partially known environments, primarily because of the tight dynamical couplings. A particular challenge is associated with inferring unknown obstacles using implicit communications when the robots have only partial knowledge of their environment, i.e. they use local sensors with limited field of view.

[1]MPC Lab, UC Berkeley, USA. [2]School of EECS, KTH, Stockholm, Sweden; E-mails: {monimoyb, y.stuerz, conradholda, fborrelli}@berkeley.edu, kallej@kth.se

We consider the task of two robots collaboratively transporting an object, constraining the robots' inputs to comply with the object's physical constraints. We consider no explicit communication, so the local environment information and the control actions are not shared between the robots. We solve the control design problem by using a leader-follower strategy with the leader using a predictive control and the follower using a simple controller, known to the leader. With this schema, the leader can solve the collaborative transportation task, with the help of the follower, while building a map of its unknown obstacles. Such a map is obtained by estimating the follower's inputs to infer missing local information about the environment sensed by the follower. This extends the work of [29], [30] to two-robot problems. Our key contributions can be summarized as follows:

1) We propose a leader-follower strategy for two robots collaboratively transporting an object in a partially known environment with obstacles. The leader solves an MPC problem based on its known set of obstacles and plans a trajectory to reach the target position, while avoiding collisions for the whole system (i.e., the two robots combined with the object to be transported).
2) We present a simple control policy for the follower that is reactive to obstacles detected by the follower (and possibly undetected by the leader). This follower control policy is designed so that it allows the leader to infer the position of obstacles not directly sensed.
3) Motivated by [15], we introduce a strategy for allowing leader-follower role switches during the task. We present a detailed numerical example of two point robots transporting a rigid rod in an initially unknown environment. On this example our proposed approach allows the leader's MPC controller to learn the undetected obstacles and successfully complete the task, with the leader-follower roles appropriately switched.

Control design with three or more robots is not addressed in this work. In order to estimate the inputs of the other robot, we assume each robot can estimate the states of the joint system, i.e., the two robots with the object to be transported. For the considered example of two point robots transporting a rigid rod, this estimation is done with measurements of robots' own positions and the rod orientation. For more complicated systems, similar estimates may be obtained using additional sensors. We do not present this in this paper.

II. Problem Formulation

In this section, we formulate the collaborative obstacle avoidance problem with the leader-follower control architecture. Such a leader-follower hierarchy is common in control

design [15], [16], [31], [32]. We limit ourselves to the case of only one follower. We refer to the two robots with the object to be transported as the *joint system*.

A. Environment Constraints

Let the environment be defined by the set $\mathcal{X}$. We assume obstacles are static, although the proposed framework can be extended to dynamic obstacles. Let the set of obstacles be denoted by $\mathcal{O}$. Therefore, the safe set for the joint system is given by $\mathcal{S} = \mathcal{X} \setminus \mathcal{O}$. At the beginning of the task, we assume that the robots do not have any prior information about the environment. During the control task, the robots detect obstacles and store their positions. At any time step t, let the set of obstacle constraints known to the leader and the follower (detected at t and stored until t) be denoted by $\mathcal{C}_{l,t}$ and $\mathcal{C}_{f,t}$, respectively. We denote:

$$\mathcal{C}_{l,t} \cup \mathcal{C}_{f,t} = \mathcal{O}_t, \text{ with } \mathcal{O}_{t-T_s} \subseteq \mathcal{O}_t, \ \forall t \leq T,$$

where T_s is the sampling time of both the leader and the follower robot controllers (defined next in Section II-B), and $T \gg T_s$ is the task duration.

B. System Modeling

We consider that the leader and the follower robots transport the same object as they move. The state space equation of the joint system is of the form:

$$S_{t+T_s} = f(S_t, u_t, v_t), \tag{1}$$

where $S_t \in \mathbb{R}^d$ is the joint system state, $u_t \in \mathbb{R}^m$ is the input of the leader and $v_t \in \mathbb{R}^p$ is the input of the follower at time step t, and $f(\cdot,\cdot,\cdot)$ is any nonlinear map.

Remark 1: In general the states S_t contain the positions and velocities of the center of masses of the leader, the follower and the object being transported.

A block diagram of the joint system is shown in Fig. 1, where the red and the blue parts indicate the operations carried out by the leader and the follower, respectively. We consider

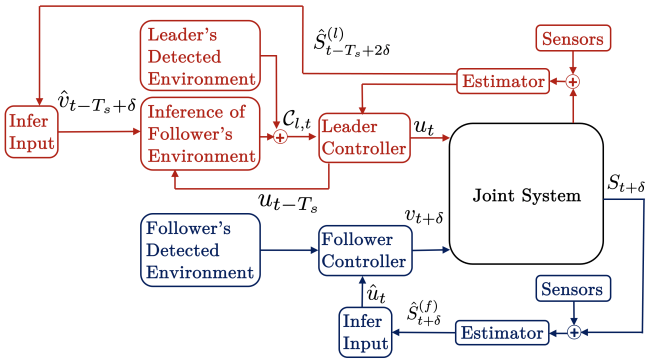

Fig. 1: Block diagram of the joint system with leader follower controllers.

the case where the leader does not have full information of all the detected obstacles in $\mathcal{O}_t$, i.e., $\mathcal{C}_{l,t} \subset \mathcal{O}_t$. We further consider that no explicit communication between the leader and the follower is available. Similar to [15]–[23], we enable both the agents to infer each other's inputs as "implicit" communication. We first make the following assumption.

Assumption 1: The leader and the follower can estimate the joint system states S_t at all time steps.

We introduce the following notation: let $(\cdot)^{(j)}_t$ denote the value of the quantity $(\cdot)_t$ as inferred by robot $j \in \{l, f\}$. The leader and the follower's estimates of S_t are thus denoted by $\hat{S}^{(l)}_t$ and $\hat{S}^{(f)}_t$, respectively. We denote the coordinates of the center of mass of the follower by $R_t = [X_{f,t}, Y_{f,t}]^\top$, and the leader/follower estimates $\hat{R}^{(l/f)}_t = [\hat{X}^{(l/f)}_{f,t}, \hat{Y}^{(l/f)}_{f,t}]^\top$. Often such states are already included in $\hat{S}^{(l/f)}_t$, as pointed out in Remark 1. If they are not a part of $\hat{S}^{(l/f)}_t$, they need to be estimated as well in our control approach.

Method Outline: At time step t, the leader uses $\hat{S}^{(l)}_t$ to compute the control action u_t for the joint system to avoid its known set of obstacles $\mathcal{C}_{l,t}$. As there is no explicit communication, the follower infers the leader's inputs u_t via its state estimates, inducing a delay in the application of its inputs. That is, at time step $t + \delta$ (with a $\delta \ll T_s$), the follower uses $\hat{S}^{(f)}_{t+\delta}$ and $\hat{R}^{(f)}_{t+\delta}$ to infer $\hat{u}_t$. The follower also uses $\hat{R}^{(f)}_{t+\delta}$ to build a map of its detected obstacles, and computes $v_{t+\delta}$ as a function of u_t and these obstacles. During the inference time between t and $t + \delta$ the follower keeps applying the previous input $v_{(t-T_s)+\delta}$. The leader then infers the follower's inputs $v_{t+\delta}$ via its state estimates to learn additional obstacles. That is, at time step $(t + 2\delta)$, the leader uses $\hat{S}^{(l)}_{t+2\delta}$ to estimate $\hat{v}_{t+\delta}$, based on which it learns the position of any additional obstacles in the follower's proximity at $t + \delta$ using $\hat{R}^{(l)}_{t+\delta}$. The leader then computes updated $\mathcal{C}_{l,t+T_s}$. In the next sections, we present the controller synthesis. We discuss the effect of the time delay δ in details in Section III-C-III-D, when we distinguish between the leader and the follower applied inputs.

Remark 2: We consider that the leader and follower robots have synchronized clocks. A short discussion of non synchronized clocks is presented in [33].

For clarity of presentation in this paper, we present a specific case of model (1). Specifically, we model both the leader and the follower as point robots m_l and m_f, with global coordinates X_l, Y_l and X_f, Y_f, respectively, transporting a rigid rod, as shown in Fig. 2. The connecting

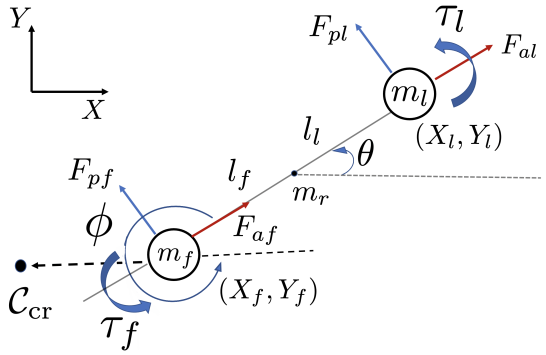

Fig. 2: Model of the joint system. The leader and the follower are point masses connected by the rigid rod. The follower reacts to a *critical obstacle* $\mathcal{C}_{\text{cr}}$, as defined in Section III-A.

rod of length $(l_l + l_f)$ has a mass m_r. Assuming the rod is of uniform density, the rod has an inertia, J_r, of $\frac{1}{12}m_r(l_l+l_f)^2$. The total mass of the joint system is therefore $m = m_r+m_l+m_f$. The total moment of inertia of the joint system is $J = J_r + m_r(\frac{l_l-l_f}{2})^2 + m_l l_l^2 + m_f l_f^2$. The leader's inputs on the rod are the axial force F_{al}, perpendicular force F_{pl}, and the torque τ_l. The corresponding follower's inputs are F_{af}, F_{pf} and τ_f. Denoting $\mathcal{T} = \frac{(-F_{pf}l_f+F_{pl}l_l+\tau_l+\tau_f)}{J}$ and define:

$$\begin{aligned} q_1 &= -(l_l \sin\theta\mathcal{T} + l_l\cos\theta\dot{\theta}^2) + \frac{1}{m}(\cos\theta(F_{al}+F_{af}) + \cdots \\ &\qquad - \sin\theta(F_{pl}+F_{pf})), \\ q_2 &= (l_l\cos\theta\mathcal{T} - l_l\sin\theta\dot{\theta}^2) + \frac{1}{m}(\sin\theta(F_{al}+F_{af})) + \cdots \\ &\qquad + \cos\theta(F_{pl}+F_{pf})). \end{aligned} \tag{2}$$

Due to the rigid coupling with the rod, the leader's position, and translational and angular velocity states are sufficient to define the evolution of the joint system. Accordingly, using (2), the state-space equation for the joint system is:

$$\begin{aligned} \dot{S}(t) &= f_c(S(t), u(t), v(t)), \\ &= \begin{bmatrix} \dot{X}_l & q_1 & \dot{Y}_l & q_2 & \dot{\theta} & \mathcal{T} \end{bmatrix}^\top, \end{aligned} \tag{3}$$

with $S(t) = [s_1, s_2, s_3, s_4, s_5, s_6]^\top$, $u(t) = [F_{al}, F_{pl}, \tau_l]^\top$, and $v(t) = [F_{af}, F_{pf}, \tau_f]^\top$ at time t, where the states are representative of variables given by:

$$s_1 = X_l, s_2 = \dot{X}_l, s_3 = Y_l, s_4 = \dot{Y}_l, s_5 = \theta, s_6 = \dot{\theta}.$$

We discretize (3) with the sampling time of T_s for both the leader and the follower to obtain its discrete time version similar to (1). Furthermore, for this specific model (3), Assumption 1 can be stated as: both the leader and the follower can estimate the leader's position, velocity, as well as the angular speed and orientation of the rod at all times. For simplicity of presentation, we consider that the robots measure their positions, velocities, and the rod's angle and angular speed. Accordingly, estimators $\hat{S}_t^{(l)}$ and $\hat{S}_t^{(f)}$ are:

$$\hat{S}_t^{(l)} = \begin{bmatrix} X_{l,t} & \dot{X}_{l,t} & Y_{l,t} & \dot{Y}_{l,t} & \theta_t & \dot{\theta}_t \end{bmatrix}^\top, \tag{4a}$$

$$\hat{S}_t^{(f)} = \begin{bmatrix} \hat{X}_{l,t}^{(f)} \\ \dot{\hat{X}}_{l,t}^{(f)} \\ \hat{Y}_{l,t}^{(f)} \\ \dot{\hat{Y}}_{l,t}^{(f)} \\ \theta_t \\ \dot{\theta}_t \end{bmatrix} = \begin{bmatrix} X_{f,t} + (l_l+l_f)\cos\theta_t \\ \dot{X}_{f,t} - (l_l+l_f)\sin\theta_t\dot{\theta}_t \\ Y_{f,t} + (l_l+l_f)\sin\theta_t \\ \dot{Y}_{f,t} + (l_l+l_f)\cos\theta_t\dot{\theta}_t \\ \theta_t \\ \dot{\theta}_t \end{bmatrix}. \tag{4b}$$

In the absence of perfect position, velocity and rod orientation measurements, one can design appropriate state observers, such as a particle filter or an extended Kalman filter to obtain their estimates, if Assumption 1 holds.

C. Input Constraints

We consider constraints on the inputs of both the leader and the follower, which are given by $u_t \in \mathcal{U}$ and $v_t \in \mathcal{V}$ for all $t \geq 0$. For our specific example in this paper, with $\bar{F}_a, \bar{F}_p, \bar{\tau} \in \mathbb{R}_+$, we consider the same box constraints:

$$\mathcal{U} = \mathcal{V} := \{w : -\begin{bmatrix} \bar{F}_a & \bar{F}_p & \bar{\tau} \end{bmatrix}^\top \leq w \leq \begin{bmatrix} \bar{F}_a & \bar{F}_p & \bar{\tau} \end{bmatrix}^\top\}. \tag{5}$$

III. Control Synthesis

We enumerate the steps involved in our leader-follower control synthesis briefly next, which constitute our *collaborative obstacle avoidance with environment learning* algorithm.

(I) At any time step t, the leader designs an MPC controller with horizon of N steps with $NT_s \ll T$ for the joint system to reach a specified target position S_{tar}, while avoiding all the stored obstacles in $\mathcal{C}_{l,t}$. This is shown in Section III-B.

(II) If there are no obstacles in its proximity, the follower uses a control strategy to support the actions of the leader. The inference of the leader actions by the follower is described in Section III-C.

(III) In the case where critical obstacles (as defined later in Definition 1) are detected by the follower, the follower applies an additional input contribution in order to avoid these critical obstacles, as we show in Section III-A.

(IV) The leader estimates the follower's applied inputs and uses this as an "implicit" communication to build a map of its possibly unseen obstacles lying in the follower's proximity. The leader then updates its set of known obstacles $\mathcal{C}_{l,t+T_s}$, as we show in Section III-D. The leader MPC problem is solved again at the next time step with the updated environment information.

We will now elaborate the above steps (I)-(IV) in the following sections.

A. Follower Policy Parameterization

In the set of all obstacles seen by the follower, we define a *critical obstacle point*, due to which the follower chooses to apply a reactive input.

Definition 1 (Critical Obstacle Points): We define a critical obstacle point at time step t as a *point* in the set of obstacles $\mathcal{C}_{f,t}$ which is within a radius of d_{cr} from the follower's center of mass. Thus, the follower computes:

$$\begin{aligned} \mathcal{C}_{\text{cr},t} = \arg\min_{c\in\mathcal{C}_{f,t}} &\|\hat{R}_t^{(f)} - c\| \\ \text{s.t., } &\|\hat{R}_t^{(f)} - c\| \leq d_{\text{cr}}, \end{aligned} \tag{6}$$

where $\|\cdot\|$ denotes the Euclidean norm.
In case of multiple critical obstacle points satisfying (6), we pick the critical obstacle point as the one that maximizes

$$\max_{c\in\mathcal{C}_{\text{cr},t}} \frac{\dot{\hat{R}}_t^{(f)} \cdot (c - \hat{R}_t^{(f)})}{\|\dot{\hat{R}}_t^{(f)}\|\|(c-\hat{R}_t^{(f)})\|},$$

that is, the one having the maximum relative velocity component towards the follower's center of mass. The inputs applied by the follower are then given by:

$$v_t = \begin{cases} f_1(u_t), & \text{if no critical obstacle point at } t, \\ f_2(u_t, d_{\text{cr}}, d_t, \phi_t) & \text{otherwise,} \end{cases} \tag{7}$$

where $f_1(\cdot)$ and $f_2(\cdot,\cdot,\cdot,\cdot)$ can be any function chosen such that $v_t \in \mathcal{V}$, u_t is the input of the leader, $d_t = \|\hat{R}_t^{(f)} - \mathcal{C}_{\mathrm{cr},t}\|$, ϕ_t is the angle between the vector connecting the follower center of mass to critical obstacle point and the follower center of mass to that of the leader, respectively. For our considered specific example, this is shown in Fig. 2.

We now make the following assumption ensuring when a critical obstacle point is seen, the follower applies a separate input, as opposed to what it would have applied otherwise.

Assumption 2: We assume in (7):

$\forall t \geq 0,\ \nexists\ u_t, d_t, \phi_t : d_t \leq d_{\mathrm{cr}},\ f_1(u_t) = f_2(u_t, d_{\mathrm{cr}}, d_t, \phi_t)$. We also make the following assumption that will be used for the leader's control synthesis in Section III-B and for learning critical obstacle points in Section III-D.

Assumption 3: We assume that the functions $f_1(\cdot)$ and $f_2(\cdot,\cdot,\cdot,\cdot)$ are known to the leader.

Assumption 3 holds true, since such basic information can be shared offline before the task begins. Otherwise, these functions can be learned offline from data.

Our specific choice of (7) in this paper is given by:

$$v_t = \begin{cases} K_2 u_t, \text{ if no critical obstacle point at } t, \text{ else} \\ K_2 u_t + K_1(d_{\mathrm{cr}} - d_t)\begin{bmatrix}\cos\phi_t & -\sin\phi_t & 0\end{bmatrix}^\top, \end{cases} \tag{8}$$

where in d_t we directly measure R_t, i.e., $\hat{R}_t^{(f)} = R_t$ (see (4b)), and the gains K_1 and K_2 known to the leader, chosen to satisfy (5). WLOG in (8), we have not included a reactive torque upon seeing critical obstacle points. Hence, only the first 2×2 sub-matrix of K_1 need to be invertible. We choose $K_2 \in [0,1)$, and

$$K_1 = \mathrm{diag}\Big(\frac{\bar{F}_a(1-K_2)}{d_{\mathrm{cr}}}, \frac{\bar{F}_p(1-K_2)}{d_{\mathrm{cr}}}, 0\Big),$$

ensuring the follower's inputs are saturated only at $d_t = 0$.

B. MPC Controller of the Leader

Using Assumption 1 and Assumption 3, the constrained finite time optimal control problem that the leader has to solve for its MPC controller synthesis at time step t is:

$$\begin{aligned} \min_{U_t}\ & \sum_{k=1}^{N}[(S_{t+kT_s|t} - S_{\mathrm{tar}})^\top Q_s (S_{t+kT_s|t} - S_{\mathrm{tar}}) + \\ & \qquad \cdots + u_{t+(k-1)T_s|t}^\top Q_i u_{t+(k-1)T_s|t}] \\ \text{s.t.},\ & S_{t+kT_s|t} = f(S_{t+(k-1)T_s|t}, u_{t+(k-1)T_s|t}, v_{t+(k-1)T_s|t}), \\ & \mathcal{B}(S_{t+kT_s|t}) \in \mathcal{X} \setminus \mathcal{C}_{l,t}, \\ & u_{t+(k-1)T_s|t} \in \mathcal{U},\ v_{t+(k-1)T_s|t} = f_1(u_{t+(k-1)T_s|t}), \\ & \forall k \in \{1,2,\ldots,N\},\ S_{t|t} = \hat{S}_t^{(l)}, \end{aligned} \tag{9}$$

where $\mathcal{B}(\cdot)$ is a set of position coordinates defining the joint leader-follower system, $U_t = \{u_{t|t}, \ldots, u_{t+(N-1)T_s|t}\}$, S_{tar} is the target state, and $Q_s, Q_i \succcurlyeq 0$ are the weight matrices. Note, in order to avoid a mixed integer formulation arising due to all possible combinations of follower's critical obstacle points in $\mathcal{C}_{l,t}$ along the prediction horizon, in (9) the leader computes the predicted $v_{k|t}$ using only $f_1(\cdot)$.

For model (3) with follower policy (8), the leader uses (4a) to estimate:

$$R_{t+kT_s|t} = \begin{bmatrix} s_{1,t+kT_s|t} - (l_f + l_r)\cos s_{5,t+kT_s|t} \\ s_{3,t+kT_s|t} - (l_f + l_r)\sin s_{5,t+kT_s|t} \end{bmatrix}, \tag{10a}$$

$$\mathcal{B}(S_{t+kT_s|t}) = \{x : \exists \alpha \in [0,1],\ x = \alpha \begin{bmatrix} s_{1,t+kT_s|t} \\ s_{3,t+kT_s|t} \end{bmatrix} + \\ + (1-\alpha) R_{t+kT_s|t}\}, \tag{10b}$$

$$v_{t+(k-1)T_s|t} = K_2 u_{t+(k-1)T_s|t} \in \mathcal{U},\ S_{t|t} = \hat{S}_t^{(l)}, \tag{10c}$$

in (9), for all $k \in \{1,2,\ldots,N\}$, with $\mathcal{U}$ from (5). Solving (9)-(10) is difficult, mostly due to the non-convexity of the imposed state constraints $\mathcal{X} \setminus \mathcal{C}_{l,t}$, and that too for all values of parameter $\alpha \in [0,1]$. Therefore, we solve an approximation to (9)-(10), as shown in [33]. After finding a solution to (9), the leader applies input

$$u_t = u_{t|t}^\star \tag{11}$$

to joint system (1) in closed-loop. Since the follower has no direct access to (11) to apply its own inputs according to (7), it estimates the leader's inputs. This is elaborated next.

C. Applying the Follower's Inputs

The follower uses $\hat{S}_t^{(f)}$ to construct an estimate $\hat{u}_t$ of the leader's inputs u_t. This inference is done in a time duration of $\delta \ll T_s$ after time step t, as introduced in Fig. 1 and Section II-B. For this inference to be feasible, we make the following sufficient assumption. Let $\hat{S}_t^{(f)} \in \mathcal{Y}_f$, $\forall t \geq 0$.

Assumption 4: We assume that the map from the set $\mathcal{U}$ to the set $\mathcal{Y}_f$ is invertible.

Assumption 4 ensures that by using its set of estimates for the leader's states, the follower has the ability to uniquely infer the input u_t applied by the leader.

Between the time steps t and $t+\delta$ the follower applies its previous inputs $v_{t-T_s+\delta}$. Afterwards, the follower applies

$$v_{t+\delta} = \begin{cases} f_1(\hat{u}_t), & \text{if no critical obstacle point at } t+\delta, \\ f_2(\hat{u}_t, d_{\mathrm{cr}}, d_{t+\delta}, \phi_{t+\delta}) & \text{otherwise}, \end{cases} \tag{12}$$

where the computation of $\hat{u}_t$ uses Assumptions 1 and 4.

For our considered system model (3), the follower's estimates of the joint system states (i.e., the leader's states) are given in (4b). This satisfies Assumption 4. The construction of the estimate $\hat{u}_t$ and the corresponding form of the follower's applied inputs

$$v_{t+\delta} = \begin{cases} K_2\hat{u}_t, \text{ if no critical obstacle point at } t+\delta, \text{ else} \\ K_2\hat{u}_t + K_1(d_{\mathrm{cr}} - d_{t+\delta})\begin{bmatrix}\cos\phi_{t+\delta} & -\sin\phi_{t+\delta} & 0\end{bmatrix}^\top \end{cases} \tag{13}$$

where $d_{t+\delta} = \|R_{t+\delta} - \mathcal{C}_{\mathrm{cr},t+\delta}\|$, are derived in detail in [33]. Here we directly measure $R_{t+\delta}$, i.e., $\hat{R}_{t+\delta}^{(f)} = R_{t+\delta}$ (see (4b)). Similar derivations can be conducted for variations of (3), e.g., the rigid connections in the system replaced by elastic spring contacts, if Assumption 4 holds.

D. Learning Critical Obstacle Points via Input Inference

The leader infers the reactive feedback of the follower in $v_{t+\delta}$ in (7) at time step $t+2\delta$. Using this, the leader's estimate of the critical obstacle point seen by the follower at time step $t+\delta$ is denoted as $\hat{\mathcal{C}}^{(l)}_{\mathrm{cr},t+\delta}$. For obtaining this estimate we first need the following assumption, along with Assumptions 1-3 stated in Section III-A. Let $\hat{S}^{(l)}_t \in \mathcal{Y}_l$, $\forall t \geq 0$.

Assumption 5: We assume that the map from the set $\mathcal{V}$ to the set $\mathcal{Y}_l$ is invertible and $f_2(\cdot, d_{\mathrm{cr}}, \cdot, \cdot)$ is an invertible function for any chosen value of the critical distance d_{cr}.

We choose function $f_2(\cdot, d_{\mathrm{cr}}, \cdot, \cdot)$ satisfying Assumption 5. Assumption 5 ensures the leader is able to uniquely infer the critical obstacle points using its estimated follower's inputs.

Satisfying Assumptions 1-3 and Assumption 5, the construction of $\hat{\mathcal{C}}^{(l)}_{\mathrm{cr},t+\delta}$ for model (3) and follower policy (8) is shown in detail in [33]. For this estimation the leader uses (4a) and computes estimates of the follower states as:

$$\begin{aligned} \hat{X}^{(l)}_{f,t+\delta} &= X_{l,t+\delta} - (l_l + l_f)\cos\theta_{t+\delta}, \\ \hat{Y}^{(l)}_{f,t+\delta} &= Y_{l,t+\delta} - (l_l + l_f)\sin\theta_{t+\delta}, \end{aligned} \tag{14}$$

and then obtains:

$$\hat{\mathcal{C}}^{(l)}_{\mathrm{cr},t+\delta} = \begin{bmatrix} \hat{X}^{(l)}_{f,t+\delta} + \hat{d}_{t+\delta}\cos(\theta_{t+\delta} - \hat{\phi}_{t+\delta}) \\ \hat{Y}^{(l)}_{f,t+\delta} + \hat{d}_{t+\delta}\sin(\theta_{t+\delta} - \hat{\phi}_{t+\delta}) \end{bmatrix}, \tag{15}$$

where $\hat{d}_{t+\delta}$ and $\hat{\phi}_{t+\delta}$ are the leader's estimate of $d_{t+\delta}$ and $\phi_{t+\delta}$, respectively. With the inferred $\hat{\mathcal{C}}^{(l)}_{\mathrm{cr},t}$, the leader then updates and uses:

$$\mathcal{C}_{l,t+T_s} = \mathcal{C}_{l,t} \cup \delta\mathcal{C}_{l,t+T_s} \cup \hat{\mathcal{C}}^{(l)}_{\mathrm{cr},t+\delta}, \tag{16}$$

where $\delta\mathcal{C}_{l,t+T_s}$ denotes the new obstacle constraints detected by the leader at the next time step. The process is then repeated from time step $(t+T_s)$ onward.

E. Leader-Follower Role Switching

Although the leader learns $\hat{\mathcal{C}}^{(l)}_{\mathrm{cr},t+\delta}$ and updates its controller, this can still lead to failure in avoiding obstacles in tight environments. For example, if the follower approaches a tight corner with multiple obstacles, the leader may not have sufficient time to generate a feasible trajectory for the joint system, as it does not directly detect the whole obstacle map from the follower and infers *only* the critical obstacle points detected by the follower. Therefore, switching the roles of the leader and the follower in these scenarios can be useful, enabling the leader to directly see all the obstacles in the tight corner. Such a role switching strategy of the leader and the follower is motivated by [15], where the roles are switched with a fixed frequency. In general, we define a time dependent role switching function for an agent as:

$$f_{\mathrm{swt}} : (x, \mathcal{C}, t) \mapsto \{0, 1\}, \tag{17}$$

where x and $\mathcal{C}$ denote the switching deciding states and obstacles of the agent, respectively, and 0 denotes no switching and 1 denotes a switch trigger.

As a specific choice for (17), we pick:

$$f_{\mathrm{swt}}(R_{t+\delta}, \mathcal{C}_{t+\delta}) = \begin{cases} 1, & \text{if } \|R_{t+\delta} - \mathcal{C}_{t+\delta}\| \leq d_{\mathrm{thr}}, \\ 0, & \text{otherwise}, \end{cases} \tag{18}$$

where d_{thr} is a chosen distance threshold value, and the follower and the leader use $\mathcal{C}_{t+\delta} = \mathcal{C}_{\mathrm{cr},t+\delta}$, $\mathcal{C}_{t+\delta} = \hat{\mathcal{C}}^{(l)}_{\mathrm{cr},t+\delta}$, and $R_{t+\delta} = \hat{R}^{(f)}_{t+\delta}$ and $R_{t+\delta} = \hat{R}^{(l)}_{t+\delta}$ obtained from (4), respectively. Having evaluated (18) at time step $t+\delta$, the agents decide the role switch trigger accordingly for control design at $t+T_s$. Since the error between $\hat{R}^{(f)}_{t+\delta}$ and $\hat{R}^{(l)}_{t+\delta}$ is zero (see (4)), the switch happens simultaneously at $t+T_s$ without any explicit communication, if the leader has an accurate estimate (15).

Remark 3 (Imperfect Estimation): If the estimate $\hat{\mathcal{C}}^{(l)}_{\mathrm{cr},t+\delta}$ has large errors, one may alternatively decide role switches with a fixed time period, similar to [15]. In such a case, the leader may not include $\hat{\mathcal{C}}^{(l)}_{\mathrm{cr},t+\delta}$ in (16), and instead include a time varying penalty in (9) based on its inferred obstacles.

IV. Numerical Experiments

We present our numerical simulations in this section. The setup is in [33], with $T_s = 0.03$s and $\delta \approx 0$. The source codes are available on: github.com/monimoyb/LeadFollowRobots. We compare the results of two alternative strategies with the one from our approach. The strategies considered are namely:

1) *Strategy 1: No Environment Learning*: The first strategy is an MPC based standard leader-follower obstacle avoidance strategy motivated by [16], [26]–[28], where the leader applies the MPC controller (9)-(11), the follower applies (8), and the leader *does not infer* any obstacle information from the follower's inputs. So the obstacles used by the leader for MPC is updated as:
$$\mathcal{C}_{l,t+T_s} = \mathcal{C}_{l,t} \cup \delta\mathcal{C}_{l,t+T_s}, \ \forall t \geq 0.$$
2) *Strategy 2: No Role Switching*: The second strategy is similar to our proposed approach, with the exception that there is *no switching* of the leader-follower roles. Such a fixed role assignment is a standard practice in the literature, e.g., [16]–[19]. As opposed to Strategy 1, here we update the leader's known set of obstacles as (16), having inferred critical obstacle points (15) using the follower's estimated inputs.
3) *Strategy 3: Environment Learning with Role Switch*: The third strategy is our proposed algorithm [33, Algorithm 1], where we learn obstacles and switch the roles of the leader and the follower using (18) and a threshold distance of $d_{\mathrm{thr}} = 0.8$ meters.

We conduct 100 trials with each of the above three strategies, with varying obstacle positions. The variations are contained in the purple regions shown on the left in Fig. 3. The shape and the size of the obstacles are unchanged, and one of their vertices is chosen uniformly in the shown regions. Successful trials are only recorded if the joint system avoids all the obstacles and the initially chosen leader robot reaches a neighborhood of radius 0.5 meters around S_{tar} within $T = 2.7$s, i.e., 90 steps. Table I summarizing the

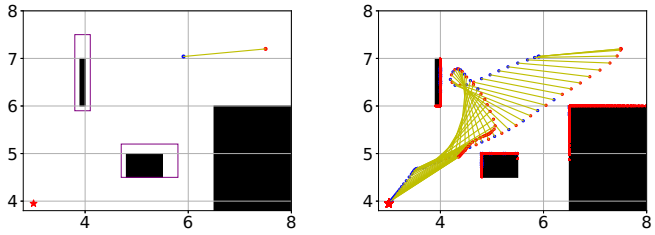

Fig. 3: Left: Zones containing varying obstacle positions with the given joint system's initial configuration. Red dot: leader, blue dot: follower, red star: S_{tar} position. Right: One example successful trajectory with the proposed approach.

TABLE I: Strategy comparison across 100 trials. CFT denotes: Collision Free Trials.

Feature	Strategy 1	Strategy 2	Strategy 3
Successful Trials (%)	0	24	86
Collision Failures (%)	100	56	14
Timed-Out Failures (%)	0	20	0
Avg. # of Steps in a CFT	N/A	87	49

results shows that our proposed algorithm outperforms the rest with the highest success rate, while reaching the target neighborhood fastest. An example of a successful trajectory with our approach is shown on the right in Fig. 3.

V. Conclusion

We proposed a leader-follower strategy for a two-robots collaborative transportation task in a partially known environment with obstacles. The leader solves an MPC problem at any given time with its known set of obstacles to plan a feasible trajectory and complete the task. The follower's policy is designed to assist the leader, but also react to additional obstacles in proximity which might be unseen to the leader. The difference between the predicted and the actual follower inputs is used by the leader to infer additional unseen environment constraints. We also propose a switching strategy for the leader-follower roles, improving the control performance in tight environments.

Acknowledgements

This project is funded by grants ONR-N00014-18-1-2833, and NSF-1931853. The project has also received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 846421.

References

[1] Z. Feng, G. Hu, Y. Sun, and J. Soon, "An overview of collaborative robotic manipulation in multi-robot systems," *Annual Reviews in Control*, 2020.

[2] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE Journal on Robotics and Automation*, vol. 3, no. 1, pp. 43–53, 1987.

[3] O. Khatib, K. Yokoi, K. Chang, D. Ruspini, R. Holmberg, and A. Casal, "Coordination and decentralized cooperation of multiple mobile manipulators," *Journal of Robotic Systems*, vol. 13, no. 11, pp. 755–764, 1996.

[4] D. Rus, B. Donald, and J. Jennings, "Moving furniture with teams of autonomous robots," in *International Conference on Intelligent Robots and Systems*, vol. 1. IEEE, 1995, pp. 235–242.

[5] P. Song and V. Kumar, "A potential field based approach to multi-robot manipulation," in *International Conference on Robotics and Automation*, vol. 2. IEEE, 2002, pp. 1217–1222.

[6] Z. Wang and V. Kumar, "Object closure and manipulation by multiple cooperating mobile robots," in *International Conference on Robotics and Automation*, vol. 1. IEEE, 2002, pp. 394–399.

[7] H. Sugie, Y. Inagaki, S. Ono, H. Aisu, and T. Unemi, "Placing objects with multiple mobile robots-mutual help using intention inference," in *International Conference on Robotics and Automation*, vol. 2. IEEE, 1995, pp. 2181–2186.

[8] B. Donald, L. Gariepy, and D. Rus, "Distributed manipulation of multiple objects using ropes," in *IEEE International Conference on Robotics and Automation*, vol. 1. IEEE, 2000, pp. 450–457.

[9] Z. Li, P. Y. Tao, S. S. Ge, M. Adams, and W. S. Wijesoma, "Robust adaptive control of cooperating mobile manipulators with relative motion," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 1, pp. 103–116, 2008.

[10] A. Franchi, A. Petitti, and A. Rizzo, "Distributed estimation of state and parameters in multiagent cooperative load manipulation," *IEEE Transactions on Control of Network Systems*, vol. 6, no. 2, pp. 690–701, 2018.

[11] A. Marino, "Distributed adaptive control of networked cooperative mobile manipulators," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 5, pp. 1646–1660, 2017.

[12] G.-B. Dai and Y.-C. Liu, "Leaderless and leader-following consensus for networked mobile manipulators with communication delays," in *Conference on Control Applications*. IEEE, 2015, pp. 1656–1661.

[13] Y. R. Sturz, A. Eichler, and R. S. Smith, "Distributed control design for heterogeneous interconnected systems," *IEEE Transactions on Automatic Control*, pp. 1–1, 2020.

[14] E. L. Zhu, Y. R. Stürz, U. Rosolia, and F. Borrelli, "Trajectory optimization for nonlinear multi-agent systems using decentralized learning model predictive control," in *Conference on Decision and Control*. IEEE, 2020, pp. 6198–6203.

[15] D. P. Losey, M. Li, J. Bohg, and D. Sadigh, "Learning from my partner's actions: Roles in decentralized robot teams," in *Conference on Robot Learning*. PMLR, 2020, pp. 752–765.

[16] Z. Wang and M. Schwager, "Kinematic multi-robot manipulation with no communication using force feedback," in *International Conference on Robotics and Automation*. IEEE, 2016, pp. 427–432.

[17] ——, "Force-amplifying n-robot transport system for cooperative planar manipulation without communication," *The International Journal of Robotics Research*, vol. 35, no. 13, pp. 1564–1586, 2016.

[18] D. J. Stilwell and J. S. Bay, "Toward the development of a material transport system using swarms of ant-like robots," in *International Conference on Robotics and Automation*. IEEE, 1993, pp. 766–771.

[19] R. Groß, F. Mondada, and M. Dorigo, "Transport of an object by six pre-attached robots interacting via physical links," in *International Conference on Robotics and Automation*. IEEE, 2006, pp. 1317–1323.

[20] Y. Aiyama, M. Hara, T. Yabuki, J. Ota, and T. Arai, "Cooperative transportation by two four-legged robots with implicit communication," *Robotics and Autonomous Systems*, vol. 29, no. 1, pp. 13–19, 1999.

[21] K. Kosuge and T. Oosumi, "Decentralized control of multiple robots handling an object," in *International Conference on Intelligent Robots and Systems*, vol. 1. IEEE, 1996, pp. 318–323.

[22] H. Takeda, Z. D. Wang, Y. Hirata, and K. Kosuge, "Load sharing algorithm for transporting an object by two mobile robots in coordination," in *International Conference on Intelligent Mechatronics and Automation*, 2004, pp. 374–378.

[23] R. Groß and M. Dorigo, "Group transport of an object to a target that only some group members may sense," in *International Conference on Parallel Problem Solving from Nature*. Springer, 2004, pp. 852–861.

[24] A. Marino and F. Pierri, "A two stage approach for distributed cooperative manipulation of an unknown object without explicit communication and unknown number of robots," *Robotics and Autonomous Systems*, vol. 103, pp. 122–133, 2018.

[25] J. Alonso-Mora, R. Knepper, R. Siegwart, and D. Rus, "Local motion planning for collaborative multi-robot manipulation of deformable objects," in *International Conference on Robotics and Automation*. IEEE, 2015, pp. 5495–5502.

[26] W. Li and R. Xiong, "Dynamical obstacle avoidance of task-constrained mobile manipulation using model predictive control," *IEEE Access*, vol. 7, pp. 88 301–88 311, 2019.

[27] M. Bharatheesha, C. Hernandez, M. Wisse, N. Giftsun, and G. Dumonteil, "Dynamic obstacle avoidance for collaborative robot applications," in *International Conference on Robotics and Automation*, 2017.

[28] O. Shorinwa and M. Schwager, "Scalable collaborative manipulation with distributed trajectory planning," in *International Conference on Intelligent Robots and Systems*. IEEE, 2020, pp. 9108–9115.

[29] A. Petrov and I. Sirota, "Control of a robot-manipulator with obstacle avoidance under little information about the environment," *IFAC Proceedings Volumes*, vol. 14, no. 2, pp. 1903–1908, 1981.

[30] G. Dumonteil, G. Manfredi, M. Devy, A. Confetti, and D. Sidobre, "Reactive planning on a collaborative robot for industrial applications," in *International Conference on Informatics in Control, Automation and Robotics*, vol. 2. IEEE, 2015, pp. 450–457.

[31] A. Bemporad and C. Rocchi, "Decentralized linear time-varying model predictive control of a formation of unmanned aerial vehicles," in *Conference on Decision and Control*. IEEE, 2011, pp. 7488–7493.

[32] G. Franzè, W. Lucia, and F. Tedesco, "A distributed model predictive control scheme for leader–follower multi-agent systems," *International Journal of Control*, vol. 91, no. 2, pp. 369–382, 2018.

[33] M. Bujarbaruah, Y. R. Stürz, C. Holda, K. H. Johansson, and F. Borrelli, "Learning environment constraints in collaborative robotics: A decentralized leader-follower approach," *arXiv preprint arXiv:2103.04460*, 2021.