

Distributed Bayesian Online Learning for Cooperative Manipulation

Pablo Budde gen. Dohmann^{1*}, Armin Lederer^{1*}, Marcel Dißemond¹, Sandra Hirche¹

Abstract—For tasks where the dynamics of multiple agents are physically coupled, the coordination between the individual agents becomes crucial, which requires exact knowledge of the interaction dynamics. This problem is typically addressed using centralized estimators, which can negatively impact the flexibility and robustness of the overall system. To overcome this shortcoming, we propose a novel distributed learning framework for the exemplary task of cooperative manipulation by applying Bayesian principles. Using only local state information each agent obtains an estimate of the object dynamics and grasp kinematics. These local estimates are combined using dynamic average consensus. Due to the strong probabilistic foundation of the method, each estimate of the object dynamics and grasp kinematics is accompanied by a measure of uncertainty, which allows to guarantee a bounded prediction error with high probability. Moreover, the Bayesian principles directly allow iterative learning with constant complexity, such that the proposed learning method can be used online in real-time applications. The effectiveness of the approach is demonstrated in a simulated cooperative manipulation task.

I. INTRODUCTION

Recent advances in communication networks allow for novel applications of distributed cooperative control approaches in multi-agent systems. Of special interest are tasks in which the dynamics of the individual agents are physically coupled, since a high degree of coordination is required for a successful completion of the goal. For such tasks typically precise knowledge of the interaction dynamics is required, which is often unavailable in real world scenarios and estimation techniques are required. In this work, we present such an estimation framework and illustrate the steps by applying it to a cooperative manipulation scenario. In cooperative manipulation the agents are physically coupled via the object and distributed control approaches have been under investigation recently [1], [2]. For such a task it is critical to know the grasp kinematics and object dynamics in order to precisely manipulate the object and avoid internal stress, which could possibly damage the object [3]. The application domain varies greatly and ranges from construction and manufacturing to service robots or search and rescue scenarios.

¹ Chair of Information-oriented Control, Technical University of Munich, Germany, pablo.dohmann, armin.lederer, marcel.dissemond, hirche@tum.de

*These authors contributed equally.

This work was supported by the European Research Council (ERC) Consolidator Grant "Safe data-driven control for human-centric systems (CO-MAN)" under grant agreement number 864686 and by the German Research Foundation (DFG) within the Joint Sino-German research project "Control and Optimization for Event-triggered Networked Autonomous Multi-agent Systems (COVEMAS)".

This problem is usually addressed by applying centralized estimators [4], [5] that use the states of all robots involved in the manipulation task. This kind of estimation framework is prone to single-point failures and inflexible regarding changes in the number of agents involved in the manipulation task. One way to overcome these shortcomings is to use a decentralized approach without any communication among the robot team. Each agent of the team calculates its own estimate of the unknown parameters. These local estimates can then be used for subsequent tasks [6]. While eliminating the requirements on the communication among the agents and introducing more flexibility, purely decentralized approaches omit the possibility of improving their local estimates by sharing those over a communication network. In order to combine the benefits of centralized and decentralized approaches, distributed estimation frameworks can be used by allowing for some communication. This can be done by locally calculating the estimates of the unknown parameters in a decentralized manner and iteratively combining them by some sort of consensus algorithm to achieve the convergence on a globally common estimate [2], [7]. Thus, the communication overhead is minimized compared to the overhead imposed by a centralized approach while maintaining the flexibility and robustness of a decentralized approach.

None of the above mentioned approaches considers information about the uncertainty of estimates, i.e. they are all equally weighted during aggregation. While this is a suitable approach when estimating parameters off-line using optimized input signals, it can significantly deteriorate the estimation performance in on-line learning scenarios with noisy measurements where input signals might not sufficiently excite all agents. Therefore, controllers relying on the estimated parameters permanently require a high robustness, since they have no information about the precision of the estimated parameters. It is well-known from centralized systems that probabilistic approaches relying on Bayesian learning methods allow to mitigate these issues, as they explicitly represent the uncertainty of estimates [8]. This allows an aggregation of learned parameters weighted by their uncertainty [9], [10], such that poor local estimates are oppressed in the overall aggregation. Moreover, a probabilistic representation exhibits many advantages in control, e.g., allowing to adapt control parameters to the uncertainty for ensuring stability [11], achieving cautiousness in control [12], or steering the system to regions with informative data [13].

In this work, we propose a Bayesian distributed learning framework for estimating the object dynamics and grasp kinematics. By applying Bayesian inference to obtain the local estimates, each estimated parameter is accompanied by

a measure of uncertainty. Moreover, the use of Bayesian principles readily allows iterative learning, such that the estimation framework can be used online in real-time applications. The estimated parameters and their uncertainties are propagated through a communication network using dynamic average consensus, such that an uncertainty-aware, distributed aggregation of the local estimates is achieved. The efficacy of the proposed framework is demonstrated in a simulation of a cooperative manipulation task.

The remainder of this work is structured as follows. The exact problem is formulated in Sec. II. The theory substantiating the main contributions of this work is given in Sec. III and evaluated in numerical simulations in Sec. IV. Sec. V briefly concludes the proposed contributions and presents future work. **Notation:** The matrices $\mathbf{0}_n$, \mathbf{I}_n and $\mathbf{1}_n$ denote the n -dimensional identity and zero matrix and vector with a 1 in each element, respectively. The dimension n is omitted if it is clear from context. The matrix $S(\cdot)$ denotes the skew-symmetric matrix such that $S(\mathbf{a})\mathbf{b} = \mathbf{a} \times \mathbf{b}$, $\forall \mathbf{a}, \mathbf{b} \in \mathbb{R}^3$. A superscript A^{ij} for a matrix \mathbf{A} denotes the element of the matrix in the i th row and j th column. Scalar operators applied to vectors and matrices denote an element-wise operation.

II. PROBLEM STATEMENT

In this work we consider N manipulators rigidly grasping and manipulating a common rigid object. Each agent i has a body-fixed frame $\{i\}$ attached to its tool center point. The object frame at the object's center of mass (COM) is denoted as $\{o\}$ and $\{w\}$ is the inertial world frame. If not stated otherwise by a leading superscript $^i(\cdot)$, all states are given with respect to the world frame $\{w\}$. The states $\mathbf{x}_i, \dot{\mathbf{x}}_i, \ddot{\mathbf{x}}_i$ denote the pose, velocity and acceleration of the i th endeffector, respectively. More precisely, the pose $\mathbf{x}_i = [\mathbf{p}_i^\top, \mathbf{q}_i^\top]^\top$ consists of the position $\mathbf{p}_i \in \mathbb{R}^3$ and the unit quaternion $\mathbf{q}_i = [\eta_i^\top \ \boldsymbol{\epsilon}_i^\top]^\top \in \text{Spin}(3)$, with a scalar real part η_i and imaginary vector part $\boldsymbol{\epsilon}_i$. With a slight abuse of notation the velocity is defined as $\dot{\mathbf{x}}_i = [\dot{\mathbf{p}}_i^\top \ \dot{\boldsymbol{\omega}}_i^\top]^\top$ with the angular velocity $[0 \ \dot{\boldsymbol{\omega}}_i^\top]^\top = 2 \frac{d}{dt} \mathbf{q}_i * \tilde{\mathbf{q}}_i$, where $*$ denotes the quaternion product and $\tilde{\mathbf{q}}_i$ is the quaternion inverse of \mathbf{q}_i . The acceleration is given as $\ddot{\mathbf{x}}_i = [\ddot{\mathbf{p}}_i^\top \ \dot{\boldsymbol{\omega}}_i^\top]^\top$. In the following the deviation from a desired value \mathbf{a}^d for any entity $\mathbf{a} \in \{\mathbf{p}_i, \dot{\mathbf{x}}_i, \ddot{\mathbf{x}}_i, \mathbf{h}_i\}$ is denoted as $\Delta \mathbf{a}_i = \mathbf{a}_i - \mathbf{a}_i^d$. For a quaternion \mathbf{q}_i the deviation from the desired orientation is denoted as $\Delta \mathbf{q}_i = [\Delta \eta_i, \Delta \boldsymbol{\epsilon}_i^\top]^\top = \mathbf{q}_i * \tilde{\mathbf{q}}_i^d$.

A. System Dynamics

The individual agents are modeled by the well known impedance equations as

$$\mathbf{M}_i \Delta \ddot{\mathbf{x}}_i + \mathbf{D}_i \Delta \dot{\mathbf{x}}_i + \mathbf{h}_i^K(\mathbf{x}_i, \mathbf{x}_i^d) = \Delta \mathbf{h}_i \quad (1)$$

where \mathbf{M}_i and \mathbf{D}_i denote the virtual mass and damping as

$$\mathbf{M}_i = \begin{bmatrix} m_i \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{J}_i \end{bmatrix}, \quad \mathbf{D}_i = \begin{bmatrix} d_i \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \delta_i \mathbf{I}_3 \end{bmatrix} \quad (2)$$

with design parameters $m_i, d_i, \delta_i \in \mathbb{R}_{>0}$ and inertia matrix \mathbf{J}_i . The endeffector wrench $\mathbf{h}_i = [\mathbf{f}_i^\top, \boldsymbol{\tau}_i^\top]^\top$ consists of

the forces $\mathbf{f}_i \in \mathbb{R}^3$ and torques $\boldsymbol{\tau}_i \in \mathbb{R}^3$. The vector \mathbf{h}_i^K represents the geometrically consistent stiffness [14] as

$$\mathbf{h}_i^K(\mathbf{x}_i, \mathbf{x}_i^d) = \begin{bmatrix} k_i \Delta \mathbf{p}_i \\ \kappa_i' \Delta \boldsymbol{\epsilon}_i \end{bmatrix} \quad (3)$$

with $\kappa_i' = 2\Delta\eta_i\kappa_i$ and scalar parameters $k_j, \kappa_j \in \mathbb{R}_{>0}$.

Remark 1: For the sake of exposition the gains $m_i, d_i, \delta_i, k_i, \kappa_i$ are chosen to be scalar, but the results of this work can be easily adapted to matrix valued gains by applying the same techniques.

The pose, velocity, and acceleration of the object $\mathbf{x}_o, \dot{\mathbf{x}}_o, \ddot{\mathbf{x}}_o$ are defined equivalent to the states of the agents i . We assume the object frame to be located at the object COM and thus the object dynamics are given as

$$\mathbf{M}_o \ddot{\mathbf{x}}_o + \mathbf{C}_o(\mathbf{x}_o, \dot{\mathbf{x}}_o) = \mathbf{h}_o \quad (4)$$

where \mathbf{h}_o denotes the effective object wrench. The object mass/inertia matrix is given as $\mathbf{M}_o = \text{blkdiag}(m_o \mathbf{I}_3, \mathbf{J}_o)$ and the wrench containing the gravity- and Coriolis-effects is $\mathbf{C}_o = \begin{bmatrix} -m_o \mathbf{g} \\ \boldsymbol{\omega}_o \times \mathbf{J}_o \boldsymbol{\omega}_o \end{bmatrix}$, where $m_o \in \mathbb{R}_{>0}$ and $\mathbf{J}_o \in \mathbb{R}^{3 \times 3}$ denote the object mass and the positive definite symmetric inertia matrix, respectively, and $\mathbf{g} \in \mathbb{R}^3$ is the gravity vector. Note that the inertia matrix \mathbf{J}_o in the world frame is not constant if the object rotates, which can be problematic during the estimation process. To combat this we can express \mathbf{J}_o as

$$\mathbf{J}_o = \mathbf{R}_o {}^o \mathbf{J}_o \mathbf{R}_o^T \quad (5)$$

where ${}^o \mathbf{J}_o$ is the constant inertia matrix denoted in the object frame.

B. Relative Kinematics

The relative position and orientation of any two frames $i \neq j \in \{1, \dots, N, o\}$ are defined as

$${}^j \mathbf{p}_i = \mathbf{R}_j^T (\mathbf{p}_i - \mathbf{p}_j) \quad (6)$$

$${}^j \mathbf{R}_i = \mathbf{R}_j^T \mathbf{R}_i, \quad (7)$$

where \mathbf{R}_j is the rotation matrix corresponding to the quaternion \mathbf{q}_j . Note that due to the rigidity assumption the relative position ${}^j \mathbf{p}_i$ and the orientation ${}^j \mathbf{q}_i$ are always constant. In order to highlight this fact, and due to their importance in the following results, we denote the constant distances ${}^o r_{j,i} = \mathbf{R}_o^T (\mathbf{p}_j - \mathbf{p}_i)$ and ${}^o \mathbf{r}_i = \mathbf{R}_o^T (\mathbf{p}_i - \mathbf{p}_o)$. Solving (6) for \mathbf{p}_j and differentiating, we obtain the relationship of translational velocities and accelerations between two frames as

$$\dot{\mathbf{p}}_j = \dot{\mathbf{p}}_i + S(\boldsymbol{\omega}_i) {}^j \mathbf{p}_i \quad (8)$$

$$\ddot{\mathbf{p}}_j = \ddot{\mathbf{p}}_i + \underbrace{(S(\dot{\boldsymbol{\omega}}_i) + S(\boldsymbol{\omega}_i)^2)}_{T(\boldsymbol{\omega}_i, \dot{\boldsymbol{\omega}}_i)} {}^j \mathbf{p}_i. \quad (9)$$

In addition, since the relative orientation ${}^j \mathbf{q}_i$ is constant for all $i = 1, \dots, N$ it holds that $\boldsymbol{\omega}_o = \boldsymbol{\omega}_i = \boldsymbol{\omega}_j$, $\dot{\boldsymbol{\omega}}_o = \dot{\boldsymbol{\omega}}_i = \dot{\boldsymbol{\omega}}_j$ for all $j = 1, \dots, N$.

C. Cooperative Manipulation

The object wrench \mathbf{h}_o can be related to the endeffector wrenches \mathbf{h}_i as

$$\mathbf{h}_o = \underbrace{\begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_3 & \cdots & \mathbf{I}_3 & \mathbf{0}_3 \\ S(\mathbf{r}_1) & \mathbf{I}_3 & \cdots & S(\mathbf{r}_N) & \mathbf{I}_3 \end{bmatrix}}_{=: \mathbf{G}} \bar{\mathbf{h}} \quad (10)$$

with the *grasp matrix* $\mathbf{G} \in \mathbb{R}^{6 \times 6N}$ and the vector of combined wrenches $\bar{\mathbf{h}} = -[\mathbf{h}_1^\top, \dots, \mathbf{h}_N^\top]^\top$. Similarly

$$\dot{\mathbf{x}} = \mathbf{G}^\top \dot{\mathbf{x}}_o, \quad (11)$$

holds, where $\dot{\mathbf{x}}$ denotes the concatenated vector of the individual agent velocities $\dot{\mathbf{x}}_i$. For the cooperative manipulation task, some coordination strategy is required in order to avoid internal stress in the object. A common solution [3] is obtained choosing the desired velocities according to (11) and compensating the object dynamics by setting $\mathbf{h}^d = -\mathbf{G}^+ \mathbf{h}_o^d$, with a desired object wrench \mathbf{h}_o^d compensating the object dynamics, and generalized inverse of the grasp matrix \mathbf{G}^+ . However, the grasp matrix \mathbf{G} and the compensation terms \mathbf{h}_o^d depend on the typically unknown parameters ${}^o\mathbf{r}_i, m_o, \mathbf{J}_o$ and an estimation strategy is indispensable for a successful coordination of the individual agents. While such estimation strategies exist, those are typically centralized and do not provide a measure of uncertainty.

D. Local Information and Communication

We assume that at each time instance k each agent i can measure its own state $\mathbf{z}_i^{(k)} = [\mathbf{x}_i^\top \ \dot{\mathbf{x}}_i^\top \ \ddot{\mathbf{x}}_i^\top]^\top$, which we will term the local information of agent i . We pose the following additional assumption on locally available information.

Assumption 1: The reference signals for all quantities of all agents, as well as all parameters of the dynamics (1) are known to each agent.

The availability of the desired quantities during the estimation process might seem like a strong assumption. However, typically an identification trajectory is designed before task execution and can be locally stored at each agent. While such an approach might result in internal stress on the object, this can be avoided by updating the identification strategy online using the consensus estimates presented in this work. The second part of Assumption 1 is not restrictive, since the constant parameters can be either exchanged before task execution or propagated through the network with methods similar to the ones presented in this work.

Assumption 2: The relative rotation matrices ${}^i\mathbf{R}_o$ are known by each agent.

This assumption consists of two parts. First, note that ${}^i\mathbf{R}_o = \mathbf{R}_i^\top \mathbf{R}_o = \text{const.}$ and as a result it is required that each agent has information about the initial object orientation \mathbf{R}_o . However, since the object frame can be oriented arbitrary this essentially means that the agents merely have to decide on a common initial orientation, which can be achieved via consensus algorithms on $SE(3)$ or methods similar to the ones presented in [15]. Second, we assume that each agent i knows the relative orientation ${}^j\mathbf{R}_o$ for all agents j . By

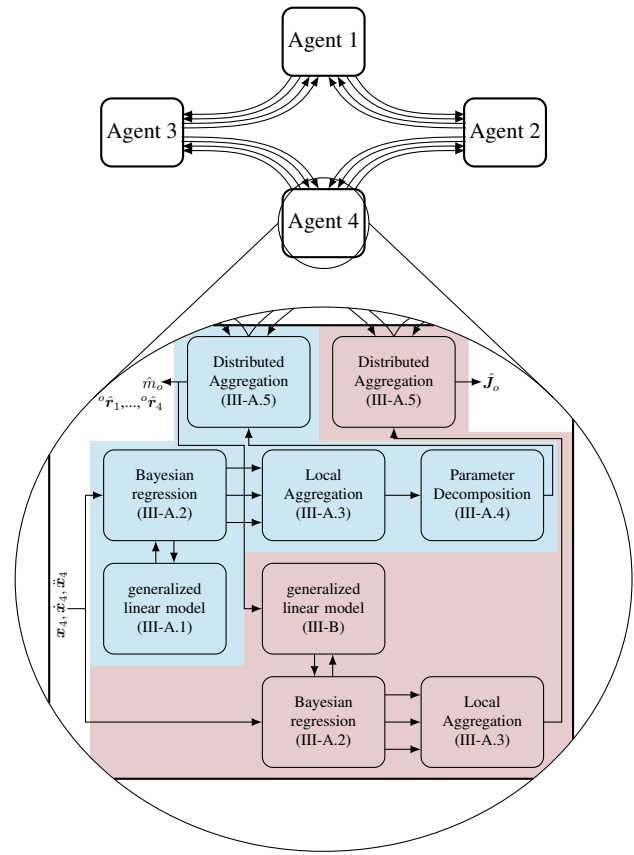


Fig. 1. Overview of the proposed distributed Bayesian learning framework. The rotational regression problem (red) depends on the estimates of the translational regression problem (blue), leading to a sequential estimation structure. For both problems we provide linear models and use Bayesian regression to obtain estimates of unknown parameters, followed by a local aggregation and parameter decomposition for the translational estimator. Finally, the estimates from all agents are aggregated in a distributed fashion.

recalling that ${}^i\mathbf{R}_o$ is constant, the relative orientations can be shared before task execution over a communication network.

Finally, we allow the agents to communicate on a communication graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V} \subseteq \{1, \dots, N\}$ is the vertex set, representing the individual agents and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the edge set, where $(i, j) \in \mathcal{E}$ if agent i and j can communicate with each other. The edges of the graph can be compactly represented through the weighted adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$, where $A^{ij} > 0$ if $(j, i) \in \mathcal{E}$. The communication network must satisfy the following properties [16].

Assumption 3: The communication graph \mathcal{G} is strongly connected and balanced, i.e., $\mathbf{A}\mathbf{1} = \mathbf{A}^\top\mathbf{1} = \mathbf{1}$. Moreover, there exists a positive constant α such that i) $A^{ii} \geq \alpha$ for all i , ii) $A^{ij} \in \{0\} \cup [\alpha, 1]$, for all i, j , iii) $\sum_{j=1}^n A^{ij} = 1$, for all i .

The goal of this work is to first estimate the parameters ${}^o\mathbf{r}_i, m_o, \mathbf{J}_o$ by using only locally available information. In a second step we want to increase the accuracy of the estimation by allowing the agents to exchange their estimates along the edges of the communication graph.

III. DISTRIBUTED BAYESIAN ONLINE LEARNING

In this section we will derive a generalized model for cooperative manipulation which is linear in the parameters $m_o {}^o \mathbf{r}_i, m_o, \mathbf{J}_o$ and depends only on information locally available at each agent. Based on this model and by applying Bayesian principles, we proceed by presenting our novel distributed estimator for the required parameters. An overview of the general framework is presented in Fig. 1. The manipulation task is split into two parts corresponding to the translational and rotational degrees of freedom, each with a separate estimator. This results in a sequential estimation procedure, where the results from the translational estimator are used in the model for the rotational one.

A. Translational Regression Problem

1) *Generalized Linear Model*: To obtain a distributed algorithm for the translational regression problem, we start by transforming all relevant equations such that only locally available information is used. Substituting the equations (6), (8), (9) in the original impedance model (1) we can use the states of agent i to express the force \mathbf{f}_j as

$$\mathbf{f}_j = m_j (\ddot{\mathbf{p}}_i + \mathbf{T}(\boldsymbol{\omega}_i, \dot{\boldsymbol{\omega}}_i) \mathbf{r}_{j,i} - \ddot{\mathbf{p}}_j^d) + d_j (\dot{\mathbf{p}}_i + S(\boldsymbol{\omega}_i) \mathbf{r}_{j,i} - \dot{\mathbf{p}}_j^d) + k_j (\mathbf{p}_i + \mathbf{r}_{j,i} - \mathbf{p}_j^d) + \mathbf{f}_j^d. \quad (12)$$

Combining this with (10) gives us an expression for the effective forces acting on the object. Finally, by transforming the object states using the equations (6), (8), (9), the left-hand side of the object dynamics (4) can also be expressed with locally available information of agent i . Separating the resulting equations into known and unknown entities yields a model of the form

$$\mathbf{y}_i = \boldsymbol{\phi}_i^\top \boldsymbol{\theta}_i \quad (13)$$

for each agent i , which is linear in the parameters

$$\begin{aligned} \boldsymbol{\theta}_i &= [\boldsymbol{\theta}_{i,1}^\top \quad \boldsymbol{\theta}_{i,2}^\top]^\top \\ \boldsymbol{\theta}_{i,1} &= [{}^o \mathbf{r}_{1,i}^\top \quad \cdots \quad {}^o \mathbf{r}_{i-1,i}^\top \quad {}^o \mathbf{r}_{i+1,i}^\top \quad \cdots \quad {}^o \mathbf{r}_{N,i}^\top]^\top \\ \boldsymbol{\theta}_{i,2} &= [m_o {}^o \mathbf{r}_i^\top \quad m_o]^\top \end{aligned} \quad (14)$$

and requires only local information, where

$$\begin{aligned} \mathbf{y}_i(\mathbf{z}_i) &= \sum_{j=1}^N [m_j \ddot{\mathbf{p}}_j^d + d_j \dot{\mathbf{p}}_j^d + k_j \mathbf{p}_j^d - \mathbf{f}_j^d] \\ &\quad - m_c \ddot{\mathbf{p}}_i - d_c \dot{\mathbf{p}}_i - k_c \mathbf{p}_i \\ \boldsymbol{\phi}_i(\mathbf{z}_i)^\top &= [\phi_{1,i}(\mathbf{z}_i) \quad \phi_{2,i}(\mathbf{z}_i) \quad \phi_{3,i}(\mathbf{z}_i)] \\ \phi_{1,i}(\mathbf{z}_i) &= \mathbf{T}(\boldsymbol{\omega}_i, \dot{\boldsymbol{\omega}}_i) \mathbf{M}_{w,i} + S(\boldsymbol{\omega}_i) \mathbf{D}_{w,i} + \mathbf{K}_{w,i} \\ \phi_{2,i}(\mathbf{z}_i) &= -\mathbf{T}(\boldsymbol{\omega}_i, \dot{\boldsymbol{\omega}}_i) \mathbf{R}_o \\ \phi_{3,i}(\mathbf{z}_i) &= \ddot{\mathbf{p}}_i - \mathbf{g}. \end{aligned} \quad (15)$$

and for any $\mathbf{B}_{w,i} \in \{\mathbf{M}_{w,i}, \mathbf{D}_{w,i}, \mathbf{K}_{w,i}\}$ and respective parameters $b_i \in \{m_i, d_i, k_i\}$ we define $\mathbf{B}_{w,i} = [b_1 \mathbf{R}_o, \dots, b_{i-1} \mathbf{R}_o, b_{i+1} \mathbf{R}_o, \dots, b_N \mathbf{R}_o]$ and $b_c = \sum_{i=1}^N b_i$. Note that in (13) the left side is an entirely determined

3-element vector of forces. The right side consists of a $3 \times (3N + 1)$ composed matrix of measurable inputs and a $(3N + 1)$ vector of unknown parameters whose values we later wish to determine.

2) *Bayesian Linear Regression*: We use Bayesian linear regression to find the estimates $\boldsymbol{\theta}_i \in \mathbb{R}^{3N+1}$ of the unknown parameters $\boldsymbol{\theta}_i$. In contrast to standard linear regression, where the estimates are obtained by minimizing the quadratic error within the linear model [17], this allows us to assess the uncertainty of the estimates and incorporate prior knowledge about the parameter values or possible noise perturbations. We place a prior probability distribution $p_i(\boldsymbol{\theta}_i) : \mathbb{R}^{3N+1} \rightarrow \mathbb{R}_{\geq 0}$ upon the parameters, which is given by

$$p_i(\boldsymbol{\theta}_i) = \mathcal{N}(\boldsymbol{\theta}_i | \boldsymbol{\mu}_i^{(0)}, \Sigma_i^{(0)}) \quad (16)$$

with the initial mean $\boldsymbol{\mu}_i^{(0)} \in \mathbb{R}^{3N+1}$ and covariance matrix $\Sigma_i^{(0)} \in \mathbb{R}^{(3N+1) \times (3N+1)}$ [8]. In order to derive closed-form expressions for inference, we assume for the moment that training targets $t_{i,m} = y_{i,m} + \varepsilon$ are output values $y_{i,m}$ perturbed by homoscedastic normally distributed i.i.d. noise $\varepsilon \sim \mathcal{N}(0, \beta^{-1})$, where the index $m = 1, 2, 3$ denotes the m th element of the three dimensional vectors \mathbf{y}_i and \mathbf{t}_i . Given a current estimate $\boldsymbol{\mu}_{i,m}^{(k)} \in \mathbb{R}^{3N+1}$, $\Sigma_{i,m}^{(k)} \in \mathbb{R}^{(3N+1) \times (3N+1)}$ and new data $\mathbf{z}_i^{(k+1)}$, the estimate can be updated by considering $p(\boldsymbol{\theta}_i | \mathbf{t}_{i,m}^{(k)})$ as prior, i.e.,

$$p(\boldsymbol{\theta}_i | \mathbf{t}_{i,m}^{(k+1)}) = \frac{p(t_{i,m}^{(k+1)} | \boldsymbol{\theta}_i, \mathbf{t}_{i,m}^{(k)}) p(\boldsymbol{\theta}_i | \mathbf{t}_{i,m}^{(k)})}{p(t_{i,m}^{(k+1)} | \mathbf{t}_{i,m}^{(k)})}, \quad (17)$$

where $\mathbf{t}_{i,m}^{(k)}$ denotes the stacked vector of output values $t_{i,m}$ at k different input instances. Since both the noise ε and the parameter vector $\boldsymbol{\theta}_i$ are Gaussian random variables, it follows from basic properties of Gaussian distributions that the posterior $p(\boldsymbol{\theta}_i | \mathbf{t}_{i,m}^{(k+1)})$ at each time step $k + 1$ is also Gaussian with mean and covariance matrix

$$\boldsymbol{\mu}_{i,m}^{(k+1)} = \Sigma_{i,m}^{(k+1)} \left((\Sigma_{i,m}^{(k)})^{-1} \boldsymbol{\mu}_{i,m}^{(k)} + \beta \phi(\mathbf{z}^{(k+1)}) t_{i,m}^{(k+1)} \right) \quad (18)$$

$$\Sigma_{i,m}^{(k+1)} = \left((\Sigma_{i,m}^{(k)})^{-1} + \beta \phi_i(\mathbf{z}^{(k+1)}) \phi_i^\top(\mathbf{z}^{(k+1)}) \right)^{-1}. \quad (19)$$

This gives an iterative update rule and allows for learning with constant update complexity.

3) *Generalized Product of Expert Aggregation*: Until now we have three estimators for each agent, which each have their individual estimate $p(\boldsymbol{\theta}_i | \mathbf{t}_{i,m}^{(k+1)})$. In order to combine these estimates without losing information about the estimates' variance or letting estimates with high uncertainty corrupt the aggregated mean, we choose a product of experts approach [9]. The key idea is to model a target probability distribution as the product of multiple densities, each provided by an expert. Experts are in this case linear Bayesian regression estimators, such that the product distribution

$$p(\boldsymbol{\theta}_i | \mathbf{t}_{i,m}^{(k)}) \propto \prod_{m=1}^3 p(\boldsymbol{\theta}_i | \mathbf{t}_{i,m}^{(k)}) \quad (20)$$

is Gaussian with mean and covariance matrix

$$\boldsymbol{\mu}_i^{(k)} = \Sigma_i^{(k)} \left(\sum_{m=1}^3 (\Sigma_{i,m}^{(k)})^{-1} \boldsymbol{\mu}_{i,m}^{(k)} \right) \quad (21)$$

$$\Sigma_i^{(k)} = \left(\sum_{m=1}^3 (\Sigma_{i,m}^{(k)})^{-1} \right)^{-1}. \quad (22)$$

However, this can be problematic since distributions without data have lower variance than the individual distributions and as a result the contribution of experts has to be normalized, which leads to the generalized product of experts as

$$\boldsymbol{\mu}_i^{(k)} = \frac{1}{3} \Sigma_i^{(k)} \left(\sum_{m=1}^3 (\Sigma_{i,m}^{(k)})^{-1} \boldsymbol{\mu}_{i,m}^{(k)} \right) \quad (23)$$

$$\Sigma_i^{(k)} = 3 \left(\sum_{m=1}^3 (\Sigma_{i,m}^{(k)})^{-1} \right)^{-1}. \quad (24)$$

Due to the structure of this aggregation, it exhibits the beneficial property that a low uncertainty in a single estimator, i.e., a covariance matrix $\Sigma_{i,m}^{(k)}$ with small entries, is sufficient to achieve an overall low uncertainty. This effect can also be observed in the mean $\boldsymbol{\mu}_i^{(k)}$, to which individual estimates $\boldsymbol{\mu}_{i,m}^{(k)}$ with high corresponding variance have a low impact.

4) *Decomposition of Regression Parameters*: For the sake of readability we denote the elements of $\boldsymbol{\mu}_i^{(k)}$ corresponding to a certain parameter $\boldsymbol{a} \in \boldsymbol{\theta}$ as $\hat{\boldsymbol{a}}$ and accordingly $\tilde{\boldsymbol{a}}$ denote the corresponding diagonal elements of the covariance matrix $\Sigma_i^{(k)}$. If not stated otherwise all quantities $\hat{\boldsymbol{a}}$ and $\tilde{\boldsymbol{a}}$ refer to agent i at time-step k . By recalling Sec. II we want to find an estimate for the parameters ${}^o\boldsymbol{r}_j, m_o, \boldsymbol{J}_o, j = 1, \dots, N$, while with the presented results, each agent i obtains estimates for $m_o, m_o {}^o\boldsymbol{r}_i, {}^o\boldsymbol{r}_j - {}^o\boldsymbol{r}_i$. The COM ${}^o\boldsymbol{r}_j$, is not a direct output of the linear Bayesian regression. When dealing with deterministic estimates, one could straightforwardly obtain ${}^o\boldsymbol{r}_i$ by dividing the estimates for $m_o {}^o\boldsymbol{r}_i$ and m_o . For Gaussian random variables, the stochastic equivalent to the deterministic division is the ratio distribution [18]. The exact distribution of the ratio of two Gaussian random variables can be calculated in closed-form under the assumption of strictly positive (or negative) mean values, but the resulting expression is rather complicated, which is prohibitive for further derivations. Therefore, we approximate the ratio distribution by a Gaussian distribution, whose mean and variance follow from a Taylor approximation of the exact ratio distribution. Considering only the marginal distributions described by the means $\widetilde{m_o {}^o\boldsymbol{r}_i}$ and the vector of diagonal elements $\widetilde{m_o {}^o\boldsymbol{r}_i}$ of the covariance matrix $\Sigma_i^{(k)}$, this yields the following identities

$$\widehat{{}^o\boldsymbol{r}_i} = \frac{\widetilde{m_o {}^o\boldsymbol{r}_i}}{\widetilde{m_o}} \quad (25)$$

$$\widetilde{{}^o\boldsymbol{r}_i} = (\widehat{{}^o\boldsymbol{r}_i})^2 \left(\frac{\widetilde{m_o {}^o\boldsymbol{r}_i}}{m_o {}^o\boldsymbol{r}_i} + \frac{\widetilde{m_o}}{\widetilde{m_o}^2} \right). \quad (26)$$

The error between the approximate Gaussian distribution with mean (25) and standard deviation (26) and the exact

ratio distribution can be shown to be bounded if $\frac{\widetilde{m_o {}^o\boldsymbol{r}_i}}{m_o {}^o\boldsymbol{r}_i}$ and $\frac{\widetilde{m_o}}{\widetilde{m_o}^2}$ are sufficiently small [18]. Since the variance of the linear Bayesian estimator decreases with a growing number of suitable data, the Gaussian distributions eventually satisfy this condition, such that modelling the exact ratio distribution using a Gaussian distribution is justified.

This estimate of ${}^o\boldsymbol{r}_i$ directly allows us to obtain estimates for ${}^o\boldsymbol{r}_j, j \neq i$. Since all considered distributions are Gaussian, ${}^o\boldsymbol{r}_j$ can be estimated by adding two Gaussian random variables, which yields a posterior Gaussian distribution with mean and variance

$$\widehat{{}^o\boldsymbol{r}_j} = \widehat{{}^o\boldsymbol{r}_i} + \widehat{{}^o\boldsymbol{r}_{j,i}} \quad (27)$$

$$\widetilde{{}^o\boldsymbol{r}_j} = \widetilde{{}^o\boldsymbol{r}_i} + \widetilde{{}^o\boldsymbol{r}_{j,i}}. \quad (28)$$

Finally, we can define the vector of means and diagonal covariance elements of the local kinematic and dynamic parameters for each agent i as

$$\hat{\boldsymbol{\mu}}_i^{(k)} = \left[\widehat{{}^o\boldsymbol{r}_1}^\top, \dots, \widehat{{}^o\boldsymbol{r}_N}^\top, \widehat{m_o} \right]^\top \quad (29)$$

$$\hat{\boldsymbol{\sigma}}_i^{(k)} = \left[\widetilde{{}^o\boldsymbol{r}_1}^\top, \dots, \widetilde{{}^o\boldsymbol{r}_N}^\top, \widetilde{m_o} \right]^\top. \quad (30)$$

5) *Distributed Model Aggregation*: It should be noted that the estimation up until this point is done by each agent individually, which results in N estimates $\hat{\boldsymbol{\mu}}_i^{(k)}, \hat{\boldsymbol{\sigma}}_i^{(k)}$. In order to combine the different predictive distributions, we can follow a generalized product of experts approach in principle again. However, the communication between agents is restricted according to a communication graph \mathcal{G} in the considered multi-agent setting, such that the direct aggregation of individual predictions is not possible. In order to mitigate this issue, we formulate the generalized product of experts aggregation as dynamic average computation, such that consensus algorithms are applicable.

In order to achieve this, we transform the local distribution parameters $\hat{\boldsymbol{\mu}}_i^{(k)}, \hat{\boldsymbol{\sigma}}_i^{(k)}$ defined in (29)-(30) using

$$\boldsymbol{\psi}_i = \left[\left(\frac{\hat{\boldsymbol{\mu}}_i^{(k)}}{\hat{\boldsymbol{\sigma}}_i^{(k)}} \right)^\top, \left(\frac{1}{\hat{\boldsymbol{\sigma}}_i^{(k)}} \right)^\top \right]^\top. \quad (31)$$

To improve to the overall estimate we take the average value of the individual estimates, computed in distributed fashion using a consensus type algorithm. Hence, we define the consensus states $\boldsymbol{\xi}_i \in \mathbb{R}^{6N+2}$ with the dynamics

$$\boldsymbol{\xi}_i^{(k+1)} = \boldsymbol{\xi}_i^{(k)} + \sum_{j \neq i}^N A^{ij} (\boldsymbol{\xi}_j^{(k)} - \boldsymbol{\xi}_i^{(k)}) + \boldsymbol{\psi}_i^{(k)} - \boldsymbol{\psi}_i^{(k-1)} \quad (32)$$

and initial state $\boldsymbol{\xi}_i^{(0)} = \boldsymbol{\psi}_i^{(0)}$, following the approach proposed in [16]. This dynamical system has been shown to exhibit a bounded consensus error under Assumption 3. Therefore, the local consensus states $\boldsymbol{\xi}_i^{(k)}$ can directly be used for computing the mean and variance of the predictive Gaussian distributions in a distributed way through

$$\begin{bmatrix} \widetilde{\boldsymbol{\mu}}_i^{(k)} \\ \widetilde{\boldsymbol{\sigma}}_i^{(k)} \end{bmatrix} = \boldsymbol{\zeta} \left(\boldsymbol{\xi}_i^{(k)} \right), \quad (33)$$

where $\zeta : \mathbb{R}^{6N+2} \rightarrow \mathbb{R}^{6N+2}$ is defined as

$$\zeta(\psi) = \begin{bmatrix} \frac{\psi_1}{\psi_{3N+2}} & \dots & \frac{\psi_{3N+1}}{\psi_{6N+2}} & \frac{1}{\psi_{3N+2}} & \dots & \frac{1}{\psi_{6N+2}} \end{bmatrix}^\top, \quad (34)$$

where ψ_m denotes the m th entry of the vector ψ . Hence, a simple dynamic average consensus algorithm in combination with a generalized product of expert aggregation allows an efficient combination of the predictive distributions resulting from local estimators.

6) *Learning Error Bound*: Due to the strong theoretical foundation of the employed methods, it is straightforward to derive a probabilistic error bound for the estimated parameters. In order to bound the estimation of Bayesian linear regression, we make the following assumption.

Assumption 4: We assume homoscedastic normally distributed errors $\varepsilon \sim \mathcal{N}(0, \beta^{-1})$ of the output \mathbf{y}_i .

Although this assumption does not reflect more realistic scenarios, where all observed states \mathbf{z} are noisy, it is only used to streamline the proof of the following theorem. In fact, it is straightforward to derive error bounds for the estimates obtained from Bayesian linear regression under more general noise distributions since the function $\phi(\cdot)$ defines a kernel. This allows the application of error bounds from Gaussian process regression, which admit, e.g., sub-Gaussian noise [19] and arbitrary bounded noise [20]. As the necessary derivations are rather cumbersome, they are omitted here due to space limitations. For bounding the error caused by the ratio distribution, we require the following assumption.

Assumption 5: For $\delta \in (0, 1)$, define $\boldsymbol{\eta}_i^{(k)} \in \mathbb{R}^{3N+1}$ as

$$\boldsymbol{\eta}_i^{(k)} = \sqrt{2 \log \left(\frac{6N(3N+1)}{\delta} \right)} \sum_{m=1}^3 \left| \frac{1}{3} \Sigma_i^{(k)} \left(\Sigma_{i,m}^{(k)} \right)^{-1} \right| \boldsymbol{\sigma}_{i,m}^{(k)}, \quad (35)$$

where $\Sigma_{i,m}^{(k)}$, $\Sigma_i^{(k)}$ are introduced in (19), (24), respectively, and $\boldsymbol{\sigma}_{i,m}^{(k)} = \text{diag}(\Sigma_{i,m}^{(k)})$. Let $\eta_{i,m_o}^{(k)}$ and $(\widehat{m}_o)_i^{(k)}$ be the elements of $\boldsymbol{\eta}_i^{(k)}$ and $\boldsymbol{\mu}_i^{(k)}$ corresponding to the parameter m_o , respectively. Then, there exists a $K \in \mathbb{N}$ such that

$$\left| (\widehat{m}_o)_i^{(k)} - \eta_{i,m_o}^{(k)} \right| > 0 \quad \forall i = 1, \dots, N, \quad \forall k \geq K. \quad (36)$$

This assumption is not restrictive since $\eta_{i,m_o}^{(k)}$ is non-increasing with respect to k , and a decrease can be guaranteed through a sufficient excitation of the system. Therefore, this condition basically requires that the trajectory \mathbf{x}^d is chosen suitably. Finally, we employ the following assumption to ensure a vanishing error of the dynamic average consensus.

Assumption 6: It holds that $\lim_{k \rightarrow \infty} 1/\widehat{\boldsymbol{\sigma}}_i^{(k)} > \mathbf{0}$.

Since we do not assume an optimal excitation signal, we cannot expect asymptotically vanishing covariance matrices $\Sigma_{i,m}^{(k)}$ in general. Therefore, $\lim_{k \rightarrow \infty} 1/\widehat{\boldsymbol{\sigma}}_i^{(k)} > \mathbf{0}$ is often satisfied in practice. Even if Assumption 6 does not hold, the static consensus error can be bounded and is usually very small, such that Theorem 1 holds approximately. Based on these assumptions, we can bound the learning error as shown in the following theorem.

Theorem 1: Consider a communication graph satisfying Assumption 3, and training data satisfying Assumptions 4-6.

Then, the error between estimated parameters $\widehat{\boldsymbol{\mu}}_i^{(k)}$ and unknown parameters $\boldsymbol{\vartheta} = [\circ \mathbf{r}_1^\top \dots \circ \mathbf{r}_N^\top m_o]^\top$ satisfies

$$\lim_{k \rightarrow \infty} P \left(|\widehat{\boldsymbol{\mu}}_i^{(k)} - \boldsymbol{\vartheta}| \leq \widehat{\boldsymbol{\eta}}^{(k)}, \quad \forall i = 1, \dots, N \right) \geq 1 - \delta \quad (37)$$

for $\widehat{\boldsymbol{\eta}}^{(k)} = \left[\left(\widehat{\boldsymbol{\eta}}_{\circ \mathbf{r}_1}^{(k)} \right)^\top \dots \left(\widehat{\boldsymbol{\eta}}_{\circ \mathbf{r}_N}^{(k)} \right)^\top \widehat{\eta}_{m_o}^{(k)} \right]^\top$ composed of

$$\widehat{\boldsymbol{\eta}}_{\circ \mathbf{r}_j}^{(k)} = \sum_{i=1}^N \frac{\left(\boldsymbol{\eta}_{\circ \mathbf{r}_i}^{(k)} + \boldsymbol{\eta}_{i, \circ \mathbf{r}_j, i}^{(k)} \right)}{\left(\widehat{\circ \mathbf{r}}_j \right)_i^{(k)} \sum_{n=1}^N \frac{1}{\left(\widehat{\circ \mathbf{r}}_j \right)_n^{(k)}}} \quad (38)$$

$$\widehat{\eta}_{m_o}^{(k)} = \sum_{i=1}^N \frac{\eta_{i, m_o}^{(k)}}{\left(\widehat{m}_o \right)_i^{(k)} \sum_{n=1}^N \frac{1}{\left(\widehat{m}_o \right)_n^{(k)}}} \quad (39)$$

$$\boldsymbol{\eta}_{\circ \mathbf{r}_i}^{(k)} = \max \left\{ \left[\frac{\left(\widehat{m}_o \circ \mathbf{r}_i \right)_i^{(k)}}{\left(\widehat{m}_o \right)_i^{(k)}} - \frac{\left(\widehat{m}_o \circ \mathbf{r}_i \right)_i^{(k)} \pm \eta_{i, m_o \circ \mathbf{r}_i}^{(k)}}{\left(\widehat{m}_o \right)_i^{(k)} \pm \eta_{i, m_o}^{(k)}} \right] \right\}, \quad (40)$$

where $\eta_{i, \circ \mathbf{r}_j, i}^{(k)}$, $\boldsymbol{\eta}_{i, m_o \circ \mathbf{r}_i}^{(k)}$ and $\eta_{i, m_o}^{(k)}$ are the elements of $\boldsymbol{\eta}_i^{(k)}$ corresponding to the parameters $\circ \mathbf{r}_j, i$, $m_o \circ \mathbf{r}_i$ and m_o , respectively.

Proof: It follows from Bayes' theorem, standard tail bounds for Gaussian distributions and the union bound that

$$P \left(\left| \boldsymbol{\mu}_{i,m}^{(k)} - \boldsymbol{\theta}_i \right| \geq \gamma \boldsymbol{\sigma}_{i,m}^{(k)} \right) \leq 2(3N+1) \exp \left(-\frac{\gamma^2}{2} \right).$$

The joint probability over all local estimators $m = 1, \dots, 3$ and all agents $i = 1, \dots, N$ can be obtained through a second application of the union bound, such that we obtain $\gamma = \sqrt{2 \log(6N(3N+1)/\delta)}$. Next, we define the weight matrices $W_{i,m}^{(k)} = \Sigma_i^{(k)} (\Sigma_{i,m}^{(k)})^{-1} / 3$. It is straightforward to see that $\sum_{m=1}^3 W_{i,m} = I$ holds, such that the estimation error after the first generalized product of experts aggregation can be bounded with probability of at least $1 - \delta$ by

$$|\boldsymbol{\mu}_i^{(k)} - \boldsymbol{\theta}_i| \leq \sum_{m=1}^3 \gamma |W_{i,m}^{(k)}| \boldsymbol{\sigma}_{i,m}^{(k)} = \boldsymbol{\eta}_i^{(k)}.$$

Due to the different treatment of estimates in the following processing steps, a case distinction is necessary. In the first case, we consider the error bound for $\widehat{\boldsymbol{\mu}}_{i,3N+1}^{(k)}$. Since this value is computed using the distributed generalized product of experts aggregation of $\boldsymbol{\mu}_{i,3N+1}^{(k)}$, the error bounds are aggregated similarly as within each agent, with the slight difference of an additional consensus error $\kappa_{i,3N+1}^{(k)}$, whose specific expression is postponed for the moment. This yields $|\widehat{\boldsymbol{\mu}}_{i,3N+1}^{(k)} - \boldsymbol{\vartheta}_{3N+1}| \leq \widehat{\eta}_{3N+1}^{(k)} + \kappa_{i,3N+1}^{(k)}$. For all other entries of $\widehat{\boldsymbol{\mu}}_i^{(k)}$, there is exactly one agent computing the estimate used in the distributed consensus based on (25), while all other agents compute the estimate through (27). Since the error bound defines a confidence interval, the quotient between (25) and the corresponding entry of $\boldsymbol{\vartheta}$ can be straightforwardly bounded using interval arithmetic [21], which results in (40) due to (36). The estimate in (27) is obtained by adding the result of (25) to the corresponding entry of $\boldsymbol{\mu}_i^{(k)}$. Thus, the error bound for (27) is obtained by adding the individual

error bounds. Due to the distributed generalized product of expert aggregation, we have in the limit $k \rightarrow \infty$ that

$$\lim_{k \rightarrow \infty} P \left(|\tilde{\boldsymbol{\mu}}_i^{(k)} - \boldsymbol{\vartheta}| \leq \hat{\boldsymbol{\eta}}^{(k)} + \boldsymbol{\kappa}_i^{(k)}, \forall i = 1, \dots, N \right) \geq 1 - \delta$$

such that it remains to prove that the consensus error $\boldsymbol{\kappa}_i^{(k)}$ asymptotically vanishes. In order to show this, we make use of [16, Corollary 3.1], which is applicable due to Assumption 3, and requires the difference between the inputs to the dynamic average consensus to vanish, i.e., $\boldsymbol{\psi}_i^{(k+1)} - \boldsymbol{\psi}_i^{(k)} \xrightarrow{1} \mathbf{0}$. This holds in the considered approach, since the posterior variance of Bayesian regression is non-increasing with respect to k and bounded from below by 0, such that the monotone convergence theorem guarantees that $\Sigma_{i,m}^{(k)} \rightarrow \bar{\Sigma}$ for some $\bar{\Sigma} \in \mathbb{R}^{(3N+1) \times (3N+1)}$. This directly implies convergence of $\boldsymbol{\mu}_{i,m}^{(k)}$, and since $1/\hat{\sigma}_i^{(k)}$ is assumed to not diverge, $\boldsymbol{\psi}_i^{(k+1)} - \boldsymbol{\psi}_i^{(k)} \rightarrow \mathbf{0}$ holds. ■

B. Rotational Regression Problem

Based on the results of the translational regression task, in this section we derive the estimator for the rotational degrees of freedom. Due to limited space and similarity in the derivation, we will omit most of the details and focus on the differences. Following similar steps as in Sec. III-A.1 we obtain the rotational model for each agent i as

$$\mathbf{y}_i^r = \boldsymbol{\phi}_i^{rT} \boldsymbol{\theta}^r \quad (41)$$

$$\begin{aligned} \mathbf{y}_i^r(\mathbf{z}) = & - \sum_{j=1}^N [S(\mathbf{r}_j) (m_j \Delta \ddot{\mathbf{p}}_j + d_j \Delta \dot{\mathbf{p}}_j + k_j \Delta \mathbf{p}_j + \mathbf{f}_j^d) \\ & + J_j \Delta \dot{\boldsymbol{\omega}}_j + \delta_j \Delta \boldsymbol{\omega}_j + \boldsymbol{\kappa}'_j \Delta \boldsymbol{\epsilon}_j + \mathbf{t}_j^d] \\ \boldsymbol{\phi}_i^{rT}(\mathbf{z}_i) = & \mathbf{V}(\mathbf{R}_o) ([\dot{\boldsymbol{\omega}}_i] + S(\boldsymbol{\omega}_i)[\boldsymbol{\omega}_i]) \\ \boldsymbol{\theta}^r = & [{}^o J_o^{11} \quad {}^o J_o^{12} \quad {}^o J_o^{13} \quad {}^o J_o^{22} \quad {}^o J_o^{23} \quad {}^o J_o^{33}]^T \\ [\dot{\boldsymbol{\omega}}] = & \begin{bmatrix} \dot{\omega}^1 & \dot{\omega}^2 & \dot{\omega}^3 & 0 & 0 & 0 \\ 0 & \dot{\omega}^1 & 0 & \dot{\omega}^2 & \dot{\omega}^3 & 0 \\ 0 & 0 & \dot{\omega}^1 & 0 & \dot{\omega}^2 & \dot{\omega}^3 \end{bmatrix} \end{aligned}$$

where \mathbf{V} is the matrix such that $[J_o^{11} \quad J_o^{12} \quad J_o^{13} \quad J_o^{22} \quad J_o^{23} \quad J_o^{33}] = \mathbf{V} \boldsymbol{\theta}^r$. Note that for the model (41) information about the unknown parameters \mathbf{r}_j and states \mathbf{z}_j of all agents is required. While it is possible to reformulate the equations using the techniques presented for the translational estimator, this would lead to a high number of unknown parameters for the regression task. In this work we follow a different approach by noticing that estimates ${}^o \mathbf{r}_i$ and ${}^o \mathbf{r}_{i,j}$ exist from the first estimator. As a result, by applying the transformations (6), (8), (9), agent i can obtain an estimate of the states \mathbf{z}_j of agent j for $i \neq j$. With this, we can reformulate

$$\begin{aligned} \mathbf{y}_i^r(\mathbf{z}_i, \tilde{\boldsymbol{\mu}}_i^{(k)}) = & - \sum_{j=1}^N [S(\hat{\mathbf{r}}_j) (m_j \widehat{\Delta \ddot{\mathbf{p}}}_j + d_j \widehat{\Delta \dot{\mathbf{p}}}_j + k_j \widehat{\Delta \mathbf{p}}_j) \\ & + J_j (\dot{\boldsymbol{\omega}}_i - \dot{\boldsymbol{\omega}}_j^d) + \delta_j (\boldsymbol{\omega}_i - \boldsymbol{\omega}_j^d) + \boldsymbol{\kappa}'_j \Delta \boldsymbol{\epsilon}_j + S(\hat{\mathbf{r}}_j) \mathbf{f}_j^d + \mathbf{t}_j^d]. \end{aligned} \quad (42)$$

Then, following the same procedure as for the translational estimator, we can use this model to perform Bayesian linear

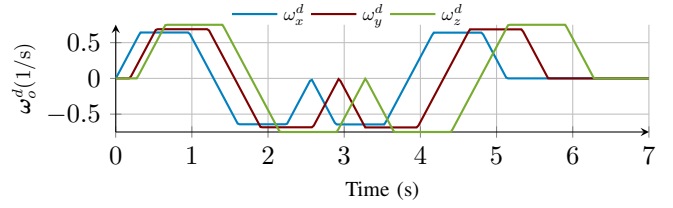


Fig. 2. Desired angular object velocity $\boldsymbol{\omega}_o^d$ around all axis (x, y, z).

regression, aggregate the individual estimates using the generalized product of experts, and perform dynamic average consensus on the resulting estimates. Note that there is no decomposition step required since the parameters $\boldsymbol{\theta}^r$ resemble the desired parameters, i.e., $\boldsymbol{\theta}^r = \boldsymbol{\vartheta}^r$. Theorem 1 can be straightforwardly adapted to this procedure by bounding the additional error caused by using $\mathbf{y}_i^r(\mathbf{z}_i, \tilde{\boldsymbol{\mu}}_i^{(k)})$ as training target instead of the true value $\mathbf{y}_i^r(\mathbf{z}_i, \boldsymbol{\vartheta})$. Since the function $\mathbf{y}_i^r(\mathbf{z}_i, \cdot)$ is Lipschitz continuous, the difference between these two targets is bounded by $\|\mathbf{y}_i^r(\mathbf{z}_i, \tilde{\boldsymbol{\mu}}_i^{(k)}) - \mathbf{y}_i^r(\mathbf{z}_i, \boldsymbol{\vartheta})\| \leq L_{yr} \|\hat{\boldsymbol{\eta}}^{(k)}\|$. This can directly be propagated through the Bayesian linear regression, i.e., the analogue of (18) for the rotational regression task. Thereby, the error caused by the targets $\mathbf{y}_i^r(\mathbf{z}_i, \tilde{\boldsymbol{\mu}}_i^{(k)})$ in the regression can be bounded by $\beta L_{yr} \|\boldsymbol{\phi}(\mathbf{z}^{(k)})\| \|\Sigma_{i,m}^{(k)}\| \|\hat{\boldsymbol{\eta}}^{(k)}\|$.

IV. NUMERICAL EVALUATION

In this section we evaluate the proposed learning framework in a simulated cooperative manipulation task, where four agents cooperatively manipulate a hollow sphere.

A. Simulation Setup

The agent dynamics in (1) are chosen with homogeneous parameters $m_i = 1, \mathbf{J}_i = 0.5 \mathbf{I}_3, d_i = 150, \delta_i = 1, k_i = 100, \boldsymbol{\kappa}_i = 0.15$ for each agent i . The agents cooperatively grasp and manipulate a hollow sphere with radius $r_o = 0.325$ and dynamics (4), where $m_o = 10, J_o = \frac{2}{3} m_o r_o^2$. The gravitational acceleration is approximated as $g = 9.81$. The initial values for the estimator are drawn from a normal distribution as $\boldsymbol{\mu}_{i,m}^{(0)} \sim \mathcal{N}(\boldsymbol{\theta}_i, \varsigma_i^2)$, where $\varsigma = [0.5 \mathbf{1}_9^T \quad 5 \mathbf{1}_3^T \quad 10 \quad \mathbf{1}_6^T]$. The initial uncertainty is chosen as $\Sigma_{i,m}^{(0)} = 0.5 \mathbf{I}$. The communication graph is given as a circular graph with adjacency matrix

$$\mathbf{A} = \begin{bmatrix} 1/3 & 1/3 & 0 & 1/3 \\ 1/3 & 1/3 & 1/3 & 0 \\ 0 & 1/3 & 1/3 & 1/3 \\ 1/3 & 0 & 1/3 & 1/3 \end{bmatrix}. \quad (43)$$

The outputs \mathbf{y}_i are corrupted by additive white noise with variance $\beta^{-1} = 2$, which is also the parameter chosen in the translational estimator. For the rotational estimator we choose $\beta_r \ll \beta$ during the first second and afterwards $\beta_r = \beta$. This is done such that only after the estimates $\boldsymbol{\theta}$ have converged, the estimation of $\boldsymbol{\theta}_r$ starts. The excitation signal consists of purely rotational desired object velocities as depicted in Fig. 2. The individual desired endeffector motion is then obtained via (11). Since for this transformation the actual values of ${}^o \mathbf{r}_i$ are

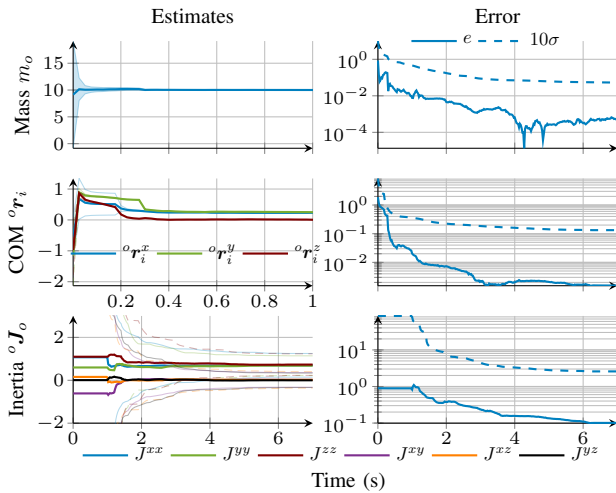


Fig. 3. Estimates (left) and estimation errors (right) including the uncertainty plotted as 10σ as shaded area. From top to bottom: mass m_o , COM ${}^o r_i$, inertia ${}^o J_o$.

required, which are unknown, they are drawn from a normal distribution $\sim \mathcal{N}({}^o r_i, 0.01)$ for the purpose of simulation.

B. Simulation Results

Due to space constraints, we only discuss the results for the first agent and note that the remaining agents yield similar results. The estimates including uncertainties are depicted in the left of Fig. 3, while the estimation error is presented in the right of Fig. 3. The estimation error is defined as the Euclidean norm of the difference between true and estimated value $e_l = \|\tilde{\mu}_l - \vartheta_l\|$, where an index $l \in \{m, r, J\}$ denotes the entries corresponding to the parameters $m_o, {}^o r_i, {}^o J_o$, respectively. In the upper two plots it can be seen that the estimates for the mass m_o and the COM ${}^o r_i$ converge towards the true values within a few time-steps, while the uncertainty decreases. After one second, the estimation errors for m_o are given as $e_m = 0.008$, which corresponds to a relative error of less than 1%, and $e_r \approx 0.02$ for the COM, corresponding to a relative error of around 6%. After one second the estimator for the inertia is activated and the estimation error decreases, as shown in the bottom plots of Fig. 3. After the simulation time of seven seconds an error of $e_J \approx 0.1$ for the inertia remains, which corresponds to roughly 8% of the actual value. Note that the estimate is accompanied with a relatively high uncertainty, such that for critical tasks control parameters could be updated to account for the uncertainty [11], which is not possible with standard approaches, which do not provide this measure of uncertainty.

V. CONCLUSION

In this paper we present a novel distributed online learning framework for cooperative manipulation using Bayesian principles. We derive a generalized linear model, which only requires locally available information. With this model the parameters are identified with Bayesian linear regression and combined with dynamic average consensus to obtain a common estimate. This allows us to provide a bound

for the prediction error with high probability and iterative learning with constant complexity, making it suitable for online learning. The approach is illustrated in a simulated cooperative manipulation setting.

REFERENCES

- [1] P. B. g. Dohmann and S. Hirche, "Distributed Control for Cooperative Manipulation With Event-Triggered Communication," *IEEE Transactions on Robotics*, vol. 36, no. 4, pp. 1038–1052, 2020.
- [2] A. Marino, "Distributed Adaptive Control of Networked Cooperative Mobile Manipulators," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 5, pp. 1646–1660, 2018.
- [3] S. Erhart and S. Hirche, "Model and Analysis of the Interaction Dynamics in Cooperative Manipulation Tasks," *IEEE Transactions on Robotics*, vol. 32, no. 3, pp. 672–683, 2016.
- [4] D. Cehajic, P. B. g. Dohmann, and S. Hirche, "Estimating unknown object dynamics in human-robot manipulation tasks," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.
- [5] F. Pierri, M. Nigro, G. Muscio, and F. Caccavale, "Cooperative Manipulation of an Unknown Object via Omnidirectional Unmanned Aerial Vehicles," *Journal of Intelligent & Robotic Systems*, vol. 100, no. 3–4, pp. 1635–1649, 2020.
- [6] A. Marino, G. Muscio, and F. Pierri, "Distributed cooperative object parameter estimation and manipulation without explicit communication," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2017.
- [7] A. Franchi, A. Petitti, and A. Rizzo, "Decentralized parameter estimation and observation for cooperative mobile manipulation of an unknown load using noisy measurements," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, 2015.
- [8] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY: Springer Science+Business Media, 2006.
- [9] Y. Cao and D. J. Fleet, "Generalized Product of Experts for Automatic and Principled Fusion of Gaussian Process Predictions," pp. 1–5, 2014. [Online]. Available: <http://arxiv.org/abs/1410.7827>
- [10] M. Deisenroth and J. W. Ng, "Distributed Gaussian Processes," in *Proceedings of the International Conference on Machine Learning*, Lille, France, 2015, pp. 1481–1490.
- [11] T. Beckers, D. Kulić, and S. Hirche, "Stable Gaussian Process based Tracking Control of Euler-Lagrange Systems," *Automatica*, vol. 103, no. 23, pp. 390–397, 2019.
- [12] L. Hewing, J. Kabzan, and M. N. Zeilinger, "Cautious Model Predictive Control using Gaussian Process Regression," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 6, pp. 2736–2743, 2020.
- [13] A. Capone, G. Noske, J. Umlauf, T. Beckers, A. Lederer, and S. Hirche, "Localized active learning of Gaussian process state space models," in *Learning for Dynamics & Control*, 2020, pp. 1–10.
- [14] F. Caccavale, P. Chiacchio, A. Marino, and L. Villani, "Six-dof impedance control of dual-arm cooperative manipulators," *IEEE/ASME Transactions on Mechatronics*, vol. 13, no. 5, pp. 576–586, 2008.
- [15] R. Tron and R. Vidal, "Distributed image-based 3-d localization of camera sensor networks," in *Proceedings of the 48th IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, 2009, pp. 901–908.
- [16] M. Zhu and S. Martínez, "Discrete-time dynamic average consensus," *Automatica*, vol. 46, no. 2, pp. 322–329, 2010.
- [17] L. Fahrmeir, T. Kneib, and S. Lang, "Regression: Modelle, Methoden und Anwendungen," Berlin, 2009.
- [18] E. Díaz-Francés and F. J. Rubio, "On the Existence of a Normal Approximation to the Distribution of the Ratio of Two Independent Normal Random Variables," *Statistical Papers*, vol. 54, no. 2, pp. 309–323, 2013.
- [19] S. R. Chowdhury and A. Gopalan, "On Kernelized Multi-armed Bandits," in *Proceedings of the International Conference on Machine Learning*, 2017, pp. 844–853.
- [20] E. T. Maddalena, P. Scharnhorst, and C. N. Jones, "Deterministic error bounds for kernel-based learning techniques under bounded noise," pp. 1–9, 2020. [Online]. Available: <https://arxiv.org/pdf/2008.04005>
- [21] G. Alefeld and G. Mayer, "Interval analysis: theory and applications," *Journal of Computational and Applied Mathematics*, vol. 121, pp. 421–464, 2000.