Collaborative Load Transportation in Microgravity Environments: Centralized and Decentralized Predictive Controllers

Sujet Phodapol[†,*], Pedro Roque[†,*], and Dimos V. Dimarogonas[†]

***Abstract*— Load transportation in micro-gravity environments is a challenging task for autonomous vehicles, but a crucial one considering the ever-increasing amount of space debris and the need for on-orbit assembly. We propose centralized and decentralized Model Predictive Controllers to transport passive loads. Realistic numerical simulations illustrate the performance of the proposed controllers for different load setpoints. While centralized controllers provide better transient performance, their computational cost makes the optimization problem intractable for more than a few agents involved in the collaborative task. On the other hand, decentralized controllers achieve faster computation time and do not suffer from the curse of dimensionality, at a small cost in performance.**

I. INTRODUCTION

The transportation of loads in microgravity environments has had applications ranging from cargo transportation [1] to autonomous on-orbit assembly [2]. As the number of satellites increases [3], [4], autonomous de-orbiting of space debris has also become of major importance for successful future space missions. Indeed, the task of autonomous load transportation is of major importance for future robotic exploration and observation missions, directly and indirectly.

Load transportation with autonomous agents is typically done using manipulators [5] or tethers [6]. Although manipulators provide more degrees of freedom for controlling a load pose, they are heavier than cabled systems and more expensive to deploy. Although tethered transportation is well-studied in the unmanned aerial vehicles (UAV) literature, such is not the case for autonomous spacecrafts.

There exists a considerable body of work around space tethers, [7]–[10]. On autonomous operations with tethered loads, [11] proposes adaptive controllers to stabilize a target, chaser, and manipulation system that can accommodate manipulators attached to tethers. In [12], a control scheme capable of stabilizing a cooperative partially failed satellite is proposed. When active control of a reel is available, [13] proposes a Robust Model Predictive Control (MPC) scheme to control the system and tether tension. For passive tethers, [14] proposes the design of Linear Quadratic Regulators and Sliding Mode controllers for active deorbiting.

Multi-agent load transportation for active debris removal seems to be far scarcer. However, the case for multi-agent missions is strong: more agents are more robust to individual failures while being able to perform transportation tasks faster due to the larger amount of collective thrust available. In [15], a path-planning scheme is used to optimize multiple agents to collaborate on a debris removal task. In [16], the authors approach the coordination scheme from a game-theoretic perspective for collaborative space-debris removal.

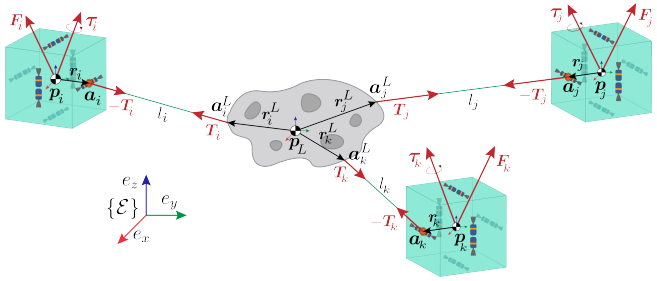


Fig. 1: System architecture with free-flyers transporting a load using tethers.

In this work, we propose the use of MPC to solve a collaborative load transportation task in microgravity environments. The controllers are deployed in a centralized and decentralized manner, and the performance is compared with respect to transient behavior and computational time. The decentralized approach does not require communication between the agents, at the expense of worse transient response when compared to centralized or distributed approaches. However, it can be implemented in scenarios where communication is difficult or costly. In short, the contributions of this work are (i) providing a generalized framework to model and calculate the tension force for the multi-agent tethered system, (ii) proposing a centralized and decentralized MPC controller to achieve a load transportation task, (iii) compare the performance of the two controllers on the same testing conditions, and (iv) provide an open-source implementation of the proposed control schemes.

The paper is divided as follows: first, we introduce the tethered system model in Section II, then we describe the system equilibrium in Section III. Sections IV and V provide the formulations for centralized and decentralized MPCs. Sections VI and VII detail the implementation of these controllers for comparative testing, and the respective results. Lastly, Section VIII concludes the paper and suggests future research directions.

Notation: Small, bold letters represent vectors. Matrices are denoted by bold, capital letters. Regular letters denote scalars. $\mathbf{a}^{\times}$ denotes the skew-symmetric matrix representation of $\mathbf{a}$, and $\mathbf{a}^T$ the transpose of $\mathbf{a}$. Calligraphic letters denote reference frames, and the basis vectors of a frame $\mathcal{A}$ are denoted $\{\mathbf{a}_x, \mathbf{a}_y, \mathbf{a}_z\}$. The inertial reference frame is generally omitted. Sets are defined in blackboard bold, $\mathbb{A}$. The weighted vector norm $\sqrt{\mathbf{x}^T\mathbf{A}\mathbf{x}}$ is denoted $\|\mathbf{x}\|_{\mathbf{A}}$.

[†]S. Phodopol, P. Roque and D. V. Dimarogonas are with the Division of Decision and Control Systems, KTH Royal Institute of Technology, Stockholm, Sweden. E-Mail: {sujet, padr, dimos}@kth.se. Sujet Phodapol and Pedro Roque contributed equally to this work.

This work was supported by the H2020 ERC Grant LEAFHOUND, the Swedish Research Council (VR), the Knut och Alice Wallenberg Foundation (KAW), and the Wallenberg AI, Autonomous Systems and Software Program (WASP) DISCOWER funded by KAW.

II. BACKGROUND

In this section, we describe the Newton-Euler dynamics of free-flyers, i.e., thruster-actuated robotic systems with holonomic motion in 6 degrees of freedom (DOF). Then, we extend the system dynamics to consider the motion with a tethered load, and finally, we present the dynamics of the full collaborative transportation system, where multiple free-flyers collaboratively transport the same load.

Consider an inertial frame $\mathcal{E}$ defined by three orthogonal axis $\{e_x, e_y, e_z\}$, and the body-fixed frame $\mathcal{B} = \{b_x, b_y, b_z\}$. We define the free-flyer's inertial position as $\boldsymbol{p}$, its velocity as $\dot{\boldsymbol{p}} = \boldsymbol{v}$, its mass as m, its rotation matrix with respect to the attitude quaternion $\boldsymbol{q}$ as $\boldsymbol{\Lambda}(\boldsymbol{q})$, its thrust as $\boldsymbol{F}$, its body-frame angular velocity as $\boldsymbol{\omega}$, its inertia as $\boldsymbol{J}$ and its body-frame torque as $\boldsymbol{\tau}$. $\boldsymbol{\Lambda}(\boldsymbol{q})$ and $\boldsymbol{\Omega}(\boldsymbol{\omega})$ are defined in [17, Eq. (92)] and [17, Eq. (108)], respectively. The dynamics of the free-flyer can be written as $\dot{\boldsymbol{p}} = \boldsymbol{v}, \quad \dot{\boldsymbol{v}} = m^{-1}(\boldsymbol{\Lambda}(\boldsymbol{q})^T \boldsymbol{F}), \quad \dot{\boldsymbol{q}} = 0.5(\boldsymbol{\Omega}(\boldsymbol{\omega}))\boldsymbol{q}, \quad \dot{\boldsymbol{\omega}} = \boldsymbol{J}^{-1}(-\boldsymbol{\omega}^{\times}\boldsymbol{J}\boldsymbol{\omega} + \boldsymbol{\tau})$. Each free-flyer has 12 thrusters, as in fig. 1. The force $\boldsymbol{F}$ and torque $\boldsymbol{\tau}$ is therefore decomposed into six input pairs $u_i, i = 1, \dots, 6$ concatenated in $\boldsymbol{u} = [u_1, \dots, u_6]^T$, where each u_i actuates a pair of thrusters (one for positive and one for negative values). We then write the thrust $\boldsymbol{F}$ and torque $\boldsymbol{\tau}$ with the thrusters' direction and arrangement matrices $\boldsymbol{D}$ and $\boldsymbol{L}$ as

$$\boldsymbol{F} = \boldsymbol{D}\boldsymbol{u} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \boldsymbol{u}, \text{ and}$$

$$\boldsymbol{\tau} = \boldsymbol{L}\boldsymbol{u} = l_{arm} \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & -1 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \end{bmatrix} \boldsymbol{u},$$

where l_{arm} is the torque length arm with respect to the free-flyer's center of mass.

Multiple Free-flyers with a Tethered Load

The system consists of $M \geq 2$ free-flyers attached with tethers to a load. When modeling the multiple rigid-bodies, we use the subscript i to refer to free-flyer i and the subscript L for the load. The tension applied by the tether on system i is written as $\boldsymbol{T}_i$. The Newton-Euler equations of agent i (1a)-(1d) and the load (1e)-(1h) can then be described as:

$$\dot{\boldsymbol{p}}_i = \boldsymbol{v}_i, \tag{1a}$$
$$\dot{\boldsymbol{v}}_i = m_i^{-1}, (\boldsymbol{\Lambda}_i^T(\boldsymbol{q}_i)\boldsymbol{F}_i - \boldsymbol{T}_i), \tag{1b}$$
$$\dot{\boldsymbol{q}}_i = \frac{1}{2}(\boldsymbol{\Omega}_i(\boldsymbol{\omega}_i))\boldsymbol{q}_i, \tag{1c}$$
$$\dot{\boldsymbol{\omega}}_i = \boldsymbol{J}_i^{-1}(-\boldsymbol{\omega}_i^{\times}\boldsymbol{J}_i\boldsymbol{\omega}_i + \boldsymbol{\tau}_i - \boldsymbol{r}_i \times (\boldsymbol{\Lambda}_i(\boldsymbol{q}_i)\boldsymbol{T}_i)), \tag{1d}$$
$$\dot{\boldsymbol{p}}_L = \boldsymbol{v}_L, \tag{1e}$$
$$\dot{\boldsymbol{v}}_L = m_L^{-1}(\boldsymbol{T}_1 + \boldsymbol{T}_2), \tag{1f}$$
$$\dot{\boldsymbol{q}}_L = \frac{1}{2}(\boldsymbol{\Omega}_L(\boldsymbol{\omega}_L))\boldsymbol{q}_L, \tag{1g}$$
$$\dot{\boldsymbol{\omega}}_L = \boldsymbol{J}_L^{-1}(-\boldsymbol{\omega}_L^{\times}\boldsymbol{J}_L\boldsymbol{\omega}_L + \sum_{i=1}^{2} \boldsymbol{r}_i^L \times (\boldsymbol{\Lambda}_L(\boldsymbol{q}_L)\boldsymbol{T}_i)), \tag{1h}$$

where $\boldsymbol{r}_i$ is the tether attach point in the body frame of agent $i = 1, ..., M$ and $\boldsymbol{r}_i^L$ is the tether attach point on the load body frame, for the tether of agent i. Fig. 1 illustrates this scenario. To facilitate the reference to these variables, we introduce the state variable $\boldsymbol{x}_i = [\boldsymbol{p}_i^T, \boldsymbol{v}_i^T, \boldsymbol{q}_i^T, \boldsymbol{\omega}_i^T]^T \in \mathbb{R}^9 \times \mathcal{SO}(3),\ i = 1, \dots, M, L$. Both the control inputs $\boldsymbol{u}_i$ and states $\boldsymbol{x}_i$ constraints are defined by polytopes, as is common in MPC frameworks [18], and defined according to

$$\mathbb{U}_i := \{u \in \mathbb{R}^6 : u_{min} \leq u_{i,[j]} \leq u_{max}, j = 1, \dots, 6\}, \tag{2}$$
$$\begin{aligned} \mathbb{X}_i := \{x \in \mathbb{R}^9 \times \mathcal{SO}(3) : p_{min} \leq \boldsymbol{p}_{i,[j]} \leq p_{max} \wedge \\ v_{min} \leq \boldsymbol{v}_{i,[j]} \leq v_{max} \wedge \omega_{min} \leq \boldsymbol{\omega}_{i,[j]} \leq \omega_{max} \\ \wedge \|\boldsymbol{q}_i\| = 1, j = x, y, z\}, \end{aligned} \tag{3}$$

where $\star_{min}, \star_{max}$ are the lower and upper bounds for each variable, respectively.

In the tethered scenario, tension forces are the internal forces that can be calculated from the states of all agents and their thrusts. The tension must respect the tether length, which needs to remain constant (i.e., all cables need to be in tension). Thus, we introduce the constraint

$$\psi_i = l_i^2 = ||(\boldsymbol{p}_i + \boldsymbol{\Lambda}(\boldsymbol{q}_i)^T \boldsymbol{r}_i) - (\boldsymbol{p}_L + \boldsymbol{\Lambda}(\boldsymbol{q}_L)^T r_i^L)||^2. \tag{4}$$

By differentiating eq. (4) twice, the tension in each cable can be written in a closed form as a function of the states and inputs in the form of

$$\begin{bmatrix} \alpha_{1,1} & \alpha_{1,2} & \cdots & \alpha_{1,M} \\ \alpha_{2,1} & \alpha_{2,2} & \cdots & \alpha_{2,M} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{M-1,1} & \alpha_{M-1,2} & \cdots & \alpha_{M-1,M} \\ \alpha_{M,1} & \alpha_{M,2} & \cdots & \alpha_{M,M} \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ \vdots \\ T_{M-1} \\ T_M \end{bmatrix} = \begin{bmatrix} \zeta_1 \\ \zeta_2 \\ \vdots \\ \zeta_{M-1} \\ \zeta_M \end{bmatrix} \tag{5}$$

where each component is defined as

$$\begin{aligned} \boldsymbol{\phi_i} &= \left(\frac{m_i + m_L}{m_i m_L}\right) \boldsymbol{I}_3 - \boldsymbol{\Lambda}_i^T \boldsymbol{r}_i^{\times} \boldsymbol{J}_i^{-1} \boldsymbol{r}_i^{\times} \boldsymbol{\Lambda}_i \\ &\quad - \boldsymbol{\Lambda}_L^T \boldsymbol{r}_i^{L\times} \boldsymbol{J}_L^{-1} \boldsymbol{r}_i^{L\times} \boldsymbol{\Lambda}_L, \\ \boldsymbol{\mu}_i &= \boldsymbol{\Lambda}_i^T \left(\frac{\boldsymbol{F}_i}{m_i} - \boldsymbol{r}_i^{\times} \boldsymbol{J}_i^{-1}(\boldsymbol{\gamma}_i + \boldsymbol{\tau}_i) + \boldsymbol{\omega}_i^{\times}\boldsymbol{\omega}_i^{\times}\boldsymbol{r}_i\right) \\ &\quad - \boldsymbol{\Lambda}_L^T \left(-\boldsymbol{r}_i^{L\times} \boldsymbol{J}_L^{-1} \boldsymbol{\gamma}_L + \boldsymbol{\omega}_L^{\times}\boldsymbol{\omega}_L^{\times}\boldsymbol{r}_i^L\right), \\ \boldsymbol{\sigma_{i,j}} &= \left(\frac{\boldsymbol{I}_3}{m_L} - \boldsymbol{\Lambda}_L^T \boldsymbol{r}_i^{L\times} \boldsymbol{J}_L^{-1} \boldsymbol{r}_j^{L\times} \boldsymbol{\Lambda}_L\right), \\ \alpha_{i,i} &= \boldsymbol{\phi}_i \hat{\boldsymbol{n}}_i \cdot \hat{\boldsymbol{n}}_i, \\ \alpha_{i,j} &= \boldsymbol{\sigma_{i,j}} \hat{\boldsymbol{n}}_j \cdot \hat{\boldsymbol{n}}_i, \\ \zeta_i &= \boldsymbol{\mu}_i \cdot \hat{\boldsymbol{n}}_i + \frac{||\cdot||_i^2}{l_i}, \end{aligned}$$

and where we introduce the shorthand notation $\boldsymbol{\Lambda}_{(\cdot)} := \boldsymbol{\Lambda}_{(\cdot)}(\boldsymbol{q}_{(\cdot)}), \quad \boldsymbol{\gamma}_{(\cdot)} := -\boldsymbol{\omega}_{(\cdot)}^{\times}\boldsymbol{J}_{(\cdot)}\boldsymbol{\omega}_{(\cdot)}, \quad ||\cdot||_i^2 := (\boldsymbol{v}_i' - \boldsymbol{v}_{s_i}) \cdot (\boldsymbol{v}_i' - \boldsymbol{v}_{s_i}), \quad \boldsymbol{n}_i := (\boldsymbol{a}_i - \boldsymbol{a}_i^L)/(\|\boldsymbol{a}_i - \boldsymbol{a}_i^L\|)$. Then, the tension force in the cable of each robot can be calculated by solving eq. (5). For a complete derivation, see [19]. We formalize our problem definition as follows:

Problem 1. *Given the system in (1), design control inputs $\boldsymbol{u}_1$ to $\boldsymbol{u}_M$ such that the load L is asymptotically stabilized to a desired state $\bar{\boldsymbol{x}}_L$, respecting the control and state constraints in (2) and (3), respectively.*

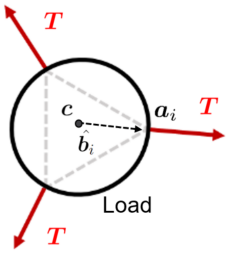


Fig. 2: The geometry representing the equilibrium configuration i defined by connecting all the anchor points.

III. SYSTEM EQUILIBRIUM

The tethered system with more than one robot has more than one equilibrium point due to redundancy. In this work, we propose a generalized method to find one specific equilibrium point which is the center of geometry created by connecting all the anchor points as shown in Fig. 2. At this point, we assume all robot attitudes to be the same as the load. Thus, we can find the center of equilibrium $\boldsymbol{c}$ of the geometry as $\boldsymbol{c} = \frac{\sum_{i=1}^{M} \boldsymbol{a}_i}{M}$ where M is the number of agents and $\boldsymbol{a}_i$ are the tether attach points in the inertial frame $\mathcal{E}$, given by $\boldsymbol{a}_i = \boldsymbol{p}_i + \boldsymbol{\Lambda}(\boldsymbol{q}_i)^T \boldsymbol{r}_i$. Next, the unit direction vector $\hat{\boldsymbol{b}}_i$ from the center of equilibrium and the positions $\boldsymbol{p}_i$ of each agent can be derived as

$$\hat{\boldsymbol{b}}_i = \frac{\boldsymbol{a}_i - \boldsymbol{c}}{||\boldsymbol{a}_i - \boldsymbol{c}||}, \;\; \boldsymbol{p}_i = \boldsymbol{a}_i + (l_i + ||\boldsymbol{r}_i||)\hat{\boldsymbol{b}}_i, \;\; i = 1, \dots, M. \tag{6}$$

IV. CENTRALIZED CONTROLLER

To solve the collaborative load transportation problem, we use MPC [18]. MPC is a Finite-Horizon Optimal Controller (FHOC) that minimizes a cost function $J(\bar{\boldsymbol{x}}_i, \boldsymbol{x}_i, \boldsymbol{u}_i)$ characterized by a desired system state $\bar{\boldsymbol{x}}_i$, the current and predicted states $\boldsymbol{x}_i$, and control input $\boldsymbol{u}_i$. The cost is minimized along a receding horizon of length N, while taking into account the discrete system model of $f(\boldsymbol{x}_i, \boldsymbol{u}_i)$. The optimization problem is constrained by state and control sets $\mathbb{X}_i$ and $\mathbb{U}_i$, as well as by state and control-dependent constraints $l_g \leq g(\boldsymbol{x}_i, \boldsymbol{u}_i) \leq u_g$, where l_g and u_g are upper and lower bounds on a nonlinear constraint defined by $g(\boldsymbol{x}_i, \boldsymbol{u}_i)$. We will appropriately design these variables to achieve the desired formation control task. Formally, we write the centralized MPC FHOC as

$$\begin{aligned} \underset{\boldsymbol{x},\boldsymbol{u}}{\textbf{minimize}} \;\; & J(\bar{\boldsymbol{x}}, \boldsymbol{x}, \boldsymbol{u}) \\ \textbf{subject to:} \;\; & \boldsymbol{x}(k+n+1|k) = f_d(\boldsymbol{x}(k), \boldsymbol{u}(k)), \\ & \boldsymbol{x}(k+n|k) \in \mathbb{X}, \\ & \boldsymbol{u}(k+n|k) \in \mathbb{U}, \\ & \boldsymbol{x}(0|0) = \boldsymbol{x}(0), \\ & n = 0, \dots, N-1, \end{aligned} \tag{7}$$

where $\boldsymbol{x} = [\boldsymbol{x}_1^T, ..., \boldsymbol{x}_M^T, \boldsymbol{x}_L^T]^T$, $\boldsymbol{u} = [\boldsymbol{u}_1^T, ..., \boldsymbol{u}_M^T]^T$, $\mathbb{X} = \mathbb{X}_1 \times ... \times \mathbb{X}_M \times \mathbb{X}_L$, $\mathbb{U} = \mathbb{U}_1 \times ... \times \mathbb{U}_M$, and $f_d(\boldsymbol{x}(k), \boldsymbol{u}(k))$ corresponds to the discretized dynamics in eq. (1) obtained through a fourth order Runge–Kutta (RK4). The desired state $\bar{\boldsymbol{x}}$ is defined as feasible setpoints for the dynamical system in eq. (1). The cost function to minimize is defined as

$$\begin{aligned} J(\bar{\boldsymbol{x}}, \boldsymbol{x}, \boldsymbol{u}) = & \sum_{n=0}^{N-1} l\big(\bar{\boldsymbol{x}}(k+n|k), \boldsymbol{x}(k+n|k), \boldsymbol{u}(k+n|k)\big) \\ & + V\big(\bar{\boldsymbol{x}}(k+n|k), \boldsymbol{x}(k+n|k)\big), \\ l\big(\bar{\boldsymbol{x}}, \boldsymbol{x}, \boldsymbol{u}\big) = & \|\boldsymbol{e}(k+n|k)\|^2_{\mathbf{Q}} + \|\boldsymbol{u}(k+n|k)\|^2_{\mathbf{R}}, \\ V\big(\bar{\boldsymbol{x}}, \boldsymbol{x}\big) = & \|\boldsymbol{e}(k+N|k)\|^2_{\mathbf{Q}_N}, \\ \boldsymbol{e}(k+n|k) = & \begin{cases} \bar{\boldsymbol{x}}(k+n|k) - \boldsymbol{x}(k+n|k), \boldsymbol{x} = [\boldsymbol{p}, \boldsymbol{v}, \boldsymbol{\omega}] \\ 1 - (\bar{\boldsymbol{x}}(k+n|k)^T \boldsymbol{x}(k+n|k))^2, \boldsymbol{x} = [\boldsymbol{q}] \end{cases} \end{aligned}$$

where $1-(\bar{\boldsymbol{q}}(k+n|k)^T \boldsymbol{q}(k+n|k))^2$ represents the quaternion distance [20]. In this scenario, and to keep control over the load, the cables must remain tensioned during the transport task. To this end, we add to eq. (7) the non-linear constraint $\boldsymbol{n}_i^T \boldsymbol{F}_i < 0$, $\boldsymbol{n}_i = \boldsymbol{a}_i - \boldsymbol{a}_i^L / \|\boldsymbol{a}_i - \boldsymbol{a}_i^L\|$, $i = 1, \dots, M$, where $\boldsymbol{a}_i^L$ is the inertial tether attached point in the load, given by $\boldsymbol{a}_i^L = \boldsymbol{p}_L + \boldsymbol{\Lambda}(\boldsymbol{q}_L)^T \boldsymbol{r}_i^L$. Such constraint ensures that the force is always applied in the half-plane opposite to the cable direction, enforcing cable tension during operation.

A. Tension Function Approximation

As seen in eq. (5), the tension variables are a highly nonlinear function of the system state. Solving in real-time the optimization problem in eq. (7) is then hard to achieve. To improve the real-time capabilities of the centralized controller, we propose another way to formulate the optimization problem. In [21], instead of explicitly computing the tension force in terms of states and input, one can reduce the nonlinearity and complexity of the original nonlinear dynamics of the system by introducing a Lagrange formulation. To simplify, the cable tension in the tethered system can be reformulated in terms of the interaction force between a group of agents and load with Lagrange multipliers $\lambda_i \in \mathcal{R}$ according to $\boldsymbol{T}_i = \lambda_i(\boldsymbol{a}_i^L - \boldsymbol{a}_i)$, $i = 1, \dots, M$, where M is the number of agents and λ_i a variable to be minimized. Moreover, we introduce an additional constraint on the second derivative of eq. (4), $\ddot{\psi}_i = 0$, implemented as inequality constraints in a discrete-time manner [21] with $\|\Psi_i(\boldsymbol{x}_i(k+n|k), \boldsymbol{u}_i(k+n|k), \lambda_i)\| \leq \epsilon_i \; i = 1, \dots, M$, for a small ϵ_i. The optimized centralized MPC formulation is then formulated as

$$\begin{aligned} \underset{\boldsymbol{u},\lambda}{\textbf{minimize}} \;\; & J(\bar{\boldsymbol{x}}, \boldsymbol{x}, \boldsymbol{u}) \\ \textbf{subject to:} \;\; & \boldsymbol{x}(k+n+1|k) = f_d(\boldsymbol{x}(n|k), \boldsymbol{u}(n|k)), \\ & \boldsymbol{x}(k+n|k) \in \mathbb{X}, \\ & \boldsymbol{u}(k+n|k) \in \mathbb{U}, \\ & |\Psi_i(k+n|k)| \leq \epsilon_i, \quad i = 1, \dots, M, \\ & \boldsymbol{n}_i^T \boldsymbol{F}_i < 0, \quad i = 1, \dots, M, \\ & \boldsymbol{T}_i = \lambda_i(\boldsymbol{a}_i^L - \boldsymbol{a}_i), \quad i = 1, \dots, M, \\ & \boldsymbol{x}(0|0) = \boldsymbol{x}(0), \\ & n = 0, ..., N-1. \end{aligned} \tag{8}$$

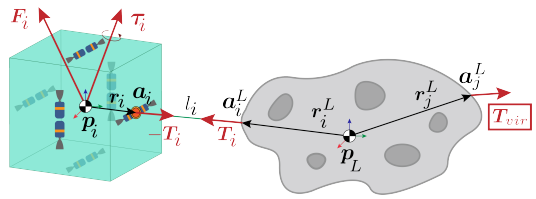


Fig. 3: A single agent performing decentralized transportation considering virtual tension forces for each neighbor.

V. DECENTRALIZED CONTROLLER

Centralized controllers suffer from the curse of dimensionality as we increase the number of agents for the optimization problem. To mitigate this problem, we propose a decentralized controller that solves a part of the optimization problem locally by approximating the behavior of the neighboring agents given the transportation task.

The proposed decentralized scheme assumes that each neighboring agent applies a constant tension $\boldsymbol{T}_{vir}$ as in fig. 3. Inspired by [22], we propose a simple method to estimate this tension using the equilibrium force. The equilibrium force is the force that each agent must apply to the load to maintain an equilibrium pose while the cables remain tensioned. This force can be approximated as a constant in the equilibrium direction $\hat{\boldsymbol{b}}$ as described in (6), which can be calculated as $\boldsymbol{T}_{vir} = \gamma\hat{\boldsymbol{b}}$, where γ is a tuning parameter. The reasoning is that if each agent expects other agents to keep the load in the equilibrium position, then this agent will solve the optimization problem as if it is the only agent that will apply the force to move the load to the desired position. As a result, all the agents collaborate to move the load to the desired position. Additionally, each agent is assumed to control only a fraction of the load, given by $m'_L = m_L/M$ and $\boldsymbol{J}'_L = \boldsymbol{J}_L/M$, where M is the number of agents. Then, the dynamics of the i^{th} subsystem can be described similarly to (1), where (1f) and (1h) are replaced by

$$\dot{\boldsymbol{v}}_L = m_L'^{-1}\left(\boldsymbol{T}_i + \sum_{j=1}^{M-1}\boldsymbol{T}_{vir_j}\right), \text{ and}$$

$$\dot{\boldsymbol{\omega}}_L = \boldsymbol{J}_L'^{-1}(-\boldsymbol{\omega}_L^{\times}\boldsymbol{J}'_L\boldsymbol{\omega}_L + \boldsymbol{r}_i^L \times (\boldsymbol{\Lambda}_L\boldsymbol{T}_i) + \sum_{j=1}^{M-1}\boldsymbol{r}_j^L \times (\boldsymbol{\Lambda}_L\boldsymbol{T}_{vir_j})).$$

The decentralized MPC controller for the agent i is then

$$\underset{\boldsymbol{u},\lambda}{\textbf{minimize}}\ J(\bar{\boldsymbol{x}}_i, \boldsymbol{x}_i, \boldsymbol{u}_i) \tag{10}$$

$$\begin{aligned}
\textbf{subject to: } & \boldsymbol{x}_i(k+n+1|k) = f_{d,dec}(\boldsymbol{x}_i(n|k), \boldsymbol{u}_i(n|k)),\\
& \boldsymbol{x}_i(k+n|k) \in \mathbb{X}_i,\\
& \boldsymbol{u}_i(k+n|k) \in \mathbb{U},\\
& |\Psi_i(k+n|k)| \leq \epsilon_i,\\
& \boldsymbol{n}_i^T\boldsymbol{F}_i < 0,\\
& \boldsymbol{T}_i = \lambda_i(\boldsymbol{a}_i^L - \boldsymbol{a}_i),\\
& \boldsymbol{x}_i(0|0) = \boldsymbol{x}_i(0),\\
& n = 0, ..., N-1,\\
& i = 1, ..., M.
\end{aligned}$$

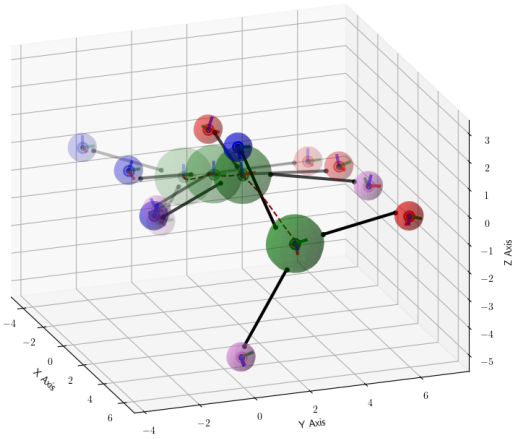


Fig. 4: Numerical Simulator in Python 3, containing a load, in green, transported by three agents, red, blue, and pink, through three different setpoints represented as cartesian frames.

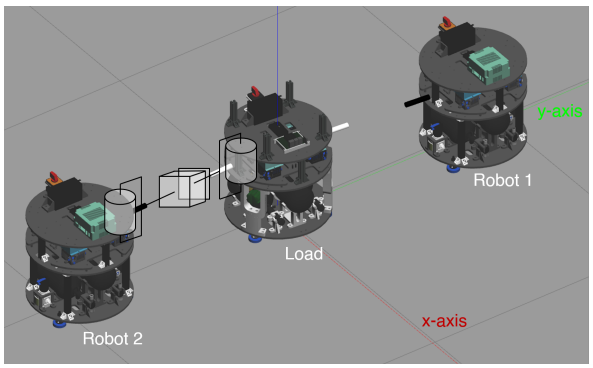


Fig. 5: Gazebo simulator displaying a prismatic and two revolute joints between each robot and load, used to emulate the cable behavior.

VI. SIMULATION SETUP

To evaluate the controllers proposed in Sections IV and V, we designed numerical experiments in Python and ROS 2. The Python 3.10 simulator is shown in fig. 4 and its source code is available in `https://github.com/pSujet/transport_mpc`, while the ROS 2 Foxy Gazebo simulator shown in fig. 5, designed to recreate the experimental conditions in the Space Robotics Laboratory at KTH, is available in `https://github.com/DISCOWER/discower_transportation`. All simulations ran in an AMD Ryzen 7 5800HS @ 2.8 GHz with 8 cores, 16 threads, and 16GB of DDR4 RAM @ 3.2 GHz. The controllers were implemented in CasADi [23].

VII. RESULTS

We ran a total of 3 tests representing the centralized controller in eq. (7), the centralized controller with the reduced model in eq. (8), and the decentralized controller in eq. (10). First, we tested the controllers in the numerical simulator with arbitrary values of mass and inertia used for both the agents and the load. This experiment displays the controllers' capability to transport the load while following the desired setpoints in 6 DoF. Then, we implemented the reduced model and decentralized controllers in ROS2 and tested them in Gazebo, which includes realistic models of the platforms available in the laboratory and a Pulse Width

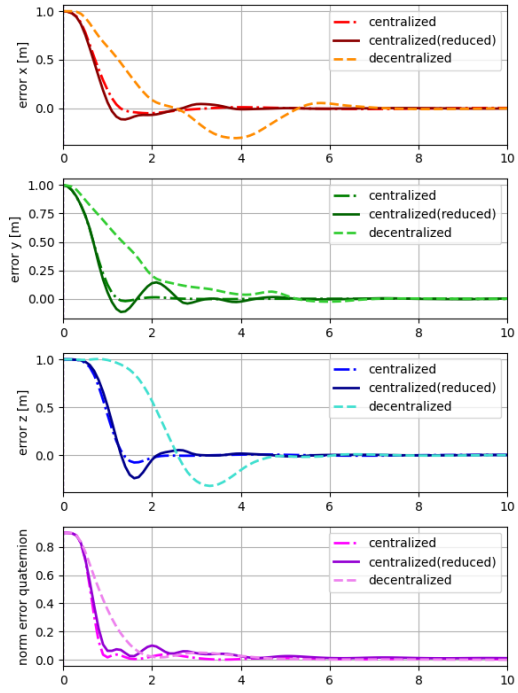


Fig. 6: Performance of the centralized controllers (explicit and implicit/reduced tension models), and the decentralized controller.

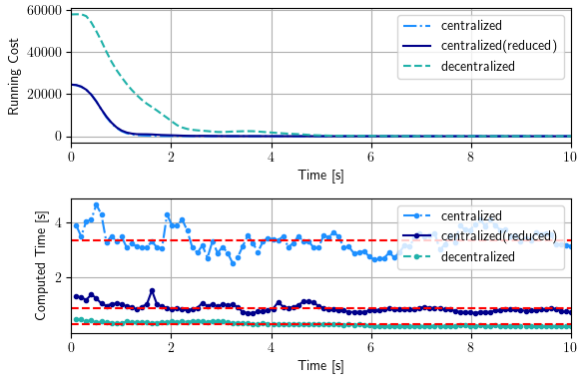


Fig. 7: MPC cost and computational time for the three proposed controllers. The average computation time for each controller is displayed by the red dashed line.

Modulation (PWM) plugin that implements the required forces in the platform body in 3 DoF.

A. Numerical Simulations

In the first experiment, we demonstrate the proposed controllers in three-dimensional space, and the results are shown in fig. 6. Three agents are the minimum number of agents to control the load in all translation and rotation axes fully. There are 52 states and 18 inputs for the system. The experiment is conducted in a numerical simulation and controlled to move the load from the origin $(0, 0, 0)$ to $(1, 1, 1)$ (1 meter in all axes) with a rotation of 45 degrees in x-y-z axis order. The target quaternion is $(0.191, 0.462, 0.191, 0.845)$, where each component corresponds to x, y, z and scale factor, respectively. As shown in Fig. 7, the controllers require 3.36s and 0.92s of average computing time, respectively. Thus, the centralized controllers cannot be implemented in real-time. The decentralized controller achieves a maximum

	Centralized	Centralized (R)	Decentralized
μ Pos. Error [m]	**0.198**	0.225	0.459
σ Pos. Error [m]	0.466	**0.461**	0.548
μ Quat. Distance [u]	**0.087**	0.110	0.124
σ Quat. Distance [u]	0.228	**0.226**	0.247
Max. CPU time [s]	4.63	1.54	**0.49**
μ CPU time [s]	3.36	0.92	**0.33**

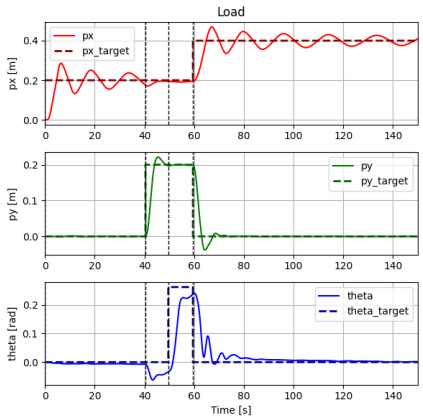
TABLE I: Statistical results corresponding to the numerical experiments performed on all controllers, where μ represents the mean and σ the standard deviation of each quantity.

computational time of 0.49s (first iteration with cold-start) and an average of 0.33s, which allows for a real-time implementation. We observe a trade-off between the centralized and decentralized control schemes on what respects performance and time-to-solution. In particular, since the decentralized case relies on the assumption that our neighbors apply the virtual torque in the equilibrium position, the transient behavior is worse than in the centralized case. The collected statistics regarding the controllers are presented in Table I where the samples were collected for 8s following the triggering of a new reference. Nonetheless, the task is still achieved with the proposed decentralized scheme, and can scale with the number of agents, while remaining real-time feasible considering a 1 Hz actuation frequency.

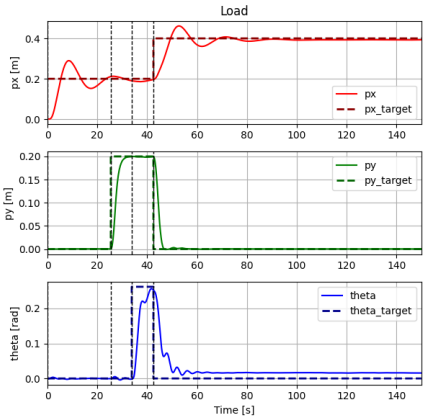
B. Gazebo Simulations

Lastly, we present the results portraying the 2D transportation task in fig. 8, using ROS2 and Gazebo. A single ROS2 node implemented the MPC problem for the centralized scenario and directly actuated each agent. In the decentralized case, however, each MPC controller was implemented as an individual node, running on the same multi-core computer as a separate thread, resembling running the experiments on two different computers onboard each vehicle. In Gazebo, a plugin was developed for PWM emulation, simulating the thrusters' behavior on the platform by transforming the desired input forces into PWM signals. The implemented controllers were the reduced model MPC (centralized), and the decentralized approaches, with 2D dynamics, and Euler discretization. The goal here is to test the controllers against a more accurate model than in the numeric simulation case.

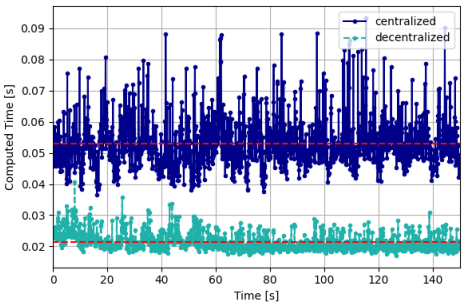
We experiment by moving the load following four desired poses: translation in the x direction to $(0.2, 0, 0)$, a translation in the y direction to $(0.2, 0.2, 0)$, a rotation of 15 degrees counterclockwise around the z-axis to pose $(0.2, 0.2, \pi/12)$, and lastly a combination of translation and rotation by moving the load to the final position $(0.4, 0, 0)$. Note that the non-optimized centralized controller could not run in real-time for two agents, and therefore it was not implemented. From fig. 8, we observe that both approaches perform well, with the centralized approach overshooting less when the combined maneuver is triggered. This is expected, as the decentralized approach approximates the model to the nominal case, contrary to the centralized one. When a combined maneuver is triggered with a large error, then the system transient will deviate from the nominal equilibrium state, leading to an overshoot. However, when we observe the computational time, the decentralized approach is almost



(a) Performance of the centralized controller. The vertical dashed line shows the trigger to the next pose.



(b) Performance of the decentralized controller. The vertical dashed line shows the trigger to the next pose.



(c) Computation cost (CPU time) of each proposed controller.

Fig. 8: Performance and computational cost for the centralized and decentralized controllers.

twice as fast as the centralized one, even in the 2-agent scenario. As the number of agents increases, this difference is expected to become more significant, as we can observe in fig. 7. The decentralized controller computational time would, by design, remain be the same.

VIII. CONCLUSIONS AND FUTURE WORK

In this work, we proposed and compared three controllers for micro-gravity tethered load transportation: two centralized and one decentralized MPC approach. We compared in numerical and realistic simulations the performance of the centralized and decentralized algorithms, concluding that the decentralized approaches outperform, in computational time, the centralized ones. However, it is also possible to observe the trade-off between the centralized and decentralized scenarios when looking at the transient performance.

For future work, we plan to extend the decentralized approach into a distributed one, taking advantage of possible real-time communication between the agents to maintain the computational time low, while improving the transient behavior. Furthermore, we are currently looking into implementing these controllers on hardware to understand how they handle the discrepancies between simulated and real dynamics, including possible disturbances.

REFERENCES

[1] M. Ekal, K. Albee, B. Coltin, R. Ventura, R. Linares, and D. W. Miller, "Online information-aware motion planning with inertial parameter learning for robotic free-flyers," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, p. 8766.

[2] S. Patane, E. R. Joyce, M. P. Snyder, and P. Shestople, "Archinaut: In-space manufacturing and assembly for next-generation space habitats," in *AIAA SPACE and astronautics forum and exposition*, 2017, p. 5227.

[3] The current state of space debris. Accessed: 2023-07-07. [Online]. Available: https://www.esa.int/Space_Safety/Space_Debris/The_current_state_of_space_debris

[4] B. Greene, "Two years of space traffic: Current trends in new payloads and debris in orbit," *Orbital Debris Quarterly News*, vol. 27, no. 2, 2023.

[5] A. Ollero, M. Tognon, A. Suarez, D. Lee, and A. Franchi, "Past, present, and future of aerial robotic manipulators," *IEEE Transactions on Robotics*, vol. 38, no. 1, p. 626, 2021.

[6] N. Michael, J. Fink, and V. Kumar, "Cooperative manipulation and transportation with aerial robots," *Autonomous Robots*, vol. 30, p. 73, 2011.

[7] M. L. Cosmo and E. C. Lorenzini, "Tethers in space handbook," Tech. Rep., 1997.

[8] E. M. Levin, *Dynamic analysis of space tether missions*. Univelt Incorporated, 2007, vol. 126.

[9] M. Cartmell and D. McKenzie, "A review of space tether research," *Progress in Aerospace sciences*, vol. 44, no. 1, p. 1, 2008.

[10] E. Papadopoulos, F. Aghili, O. Ma, and R. Lampariello, "Robotic manipulation and capture in space: A survey," *Frontiers in Robotics and AI*, p. 228, 2021.

[11] D. Wang, P. Huang, and Z. Meng, "Coordinated stabilization of tumbling targets using tethered space manipulators," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 3, p. 2420, 2015.

[12] P. Huang, Y. Lu, M. Wang, Z. Meng, Y. Zhang, and F. Zhang, "Postcapture attitude takeover control of a partially failed spacecraft with parametric uncertainties," *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 2, p. 919, 2018.

[13] B. Wang, Z. Meng, C. Jia, and P. Huang, "Reel-based tension control of tethered space robots," *IEEE transactions on aerospace and electronic systems*, vol. 56, no. 4, p. 3028, 2019.

[14] H. Linskens and E. Mooij, "Tether dynamics analysis and guidance and control design for active space-debris removal," *Journal of Guidance, Control, and Dynamics*, vol. 39, no. 6, p. 1232, 2016.

[15] J. Stuart, K. Howell, and R. Wilson, "Application of multi-agent coordination methods to the design of space debris mitigation tours," *Advances in Space Research*, vol. 57, no. 8, p. 1680, 2016.

[16] R. Klima, D. Bloembergen, R. Savani, K. Tuyls, A. Wittig, A. Sapera, and D. Izzo, "Space debris removal: Learning to cooperate and the price of anarchy," *Frontiers in Robotics and AI*, vol. 5, p. 54, 2018.

[17] N. Trawny and S. I. Roumeliotis, "Indirect kalman filter for 3d attitude estimation," *University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep*, vol. 2, 2005.

[18] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model predictive control: theory, computation, and design*. Nob Hill Publishing Madison, WI, 2017, vol. 2.

[19] S. Phodapol, "Predictive controllers for load transportation in microgravity environments," Master's thesis, KTH Royal Institute of Technology, 2023.

[20] D. Q. Huynh, "Metrics for 3D rotations: Comparison and analysis," *Journal of Mathematical Imaging and Vision*, vol. 35, no. 2, p. 155, 2009.

[21] J. Kim, R. T. Fawcett, V. R. Kamidi, A. D. Ames, and K. A. Hamed, "Layered control for cooperative locomotion of two quadrupedal robots: Centralized and distributed approaches," *IEEE Transactions on Robotics*, p. 4728, 2023.

[22] R. C. Sundin, P. Roque, and D. V. Dimarogonas, "Decentralized model predictive control for equilibrium-based collaborative uav bar transportation," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, p. 4915.

[23] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, p. 1, 2019.