2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Distributed Model Predictive Control for UAVs Collaborative Payload Transport

Jad Wehbeh[1], Shatil Rahman[1] and Inna Sharf[2]

Abstract— **We consider the problem of collaborative transport of a payload using several quadrotor vehicles. The payload is assumed to be a rigid body and is attached to the vehicles with rigid rods. The model of the system is presented and is employed to formulate a Model Predictive Controller. The centralized MPC formulation differs from others in the literature in the way the linearized model of the system is employed about a non-equilibrium state-input pair. We then present a decentralized formulation of MPC by distributing the computations among the vehicles. Simulations of both versions of the controller are carried out for a four-quadrotor system carrying out a transport maneuver of a box payload, for a cost penalizing the deviations of the vehicles from the desired trajectory and the attitude perturbations of the payload. The results confirm that the decentralized controller can yield a comparable performance to the centralized MPC implementation, for the same computation time of the two algorithms.**

I. INTRODUCTION

A. Background and Motivation

Small UAVs have witnessed increasing usage due to shifts in basic technologies contributing to making them cheaper, more user-friendly and easier to operate. This has opened up new opportunities for a number of UAV-based applications that were once infeasible, such as their use for payload transportation and delivery—a topic that has seen widespread interest from industry, start-ups and academia. Particularly, the potential uses of payload carrying UAVs for emergency response assistance and consumer package delivery have been the subject of numerous articles in the media, with successful implementations already in common practice [1].

A relevant and challenging problem that has received considerable attention from the research community is the problem of a quadrotor vehicle with a slung payload. The dynamics and control of a single-quadrotor slung-payload system have been thoroughly analyzed, and several demonstrations showcasing the transport of an object with such a system have already been performed [2], [3]. However, small UAVs, such as the majority of commercial quadrotor platforms, are typically very limited in the weight and size of payload they can carry. Moreover, they also have issues with stabilizing slung payloads while traveling to a target location [4], and can be cost-inefficient when used to transport less than their payload capacity. With this perspective, several research groups have recently started to address the problem of collaborative payload transport with multiple UAVs.

B. State of Art

In the context of collaborative transport of a slung payload with multiple quadrotors, which is the focus of the present work, several researchers have considered the dynamics, control and planning of this system. A recent comparative overview of research on 'multi-lift' rotorcraft systems is presented in [5] and includes research conducted with helicopters, quadrotors and hexacopter vehicles. The authors of [5] highlight the dearth of experimental demonstrations and the need for further research on the control of such systems. With regard to dynamics, the modeling approaches have progressed from modeling the payload as a disturbance [6], [7], thus completely omitting the coupling between the dynamics of the vehicles, to the more recent work where the payload is modeled as a rigid body [8], [9]. The approach to modeling is closely tied to the choice of control strategy for the system. Disturbance based control strategies aim to maintain UAV formation [6], [10], [11]. Point mass payload models are used for control and planning of swing free payload trajectories [12]–[14]. Full 6 DoF payload models allow to generate attitude trajectories for the payload that minimize twisting [9], [15].

Distributed model predictive control (DMPC) algorithms, such as the one presented here, solve for the optimal control inputs of multi-agent systems on different processors. The computational load is thus shared between the different agents, which communicate any required information across the local controllers. These have been studied extensively, with reviews of the subject presented in [16] and [17].

DMPC algorithms have already been explored in the context of UAVs, with a focus on online planning and obstacle avoidance. Previous work on the subject studies the behaviour of such systems under various linear [18]–[20] and nonlinear [21]–[23] MPC controllers, proposes approaches to controlling systems with slung load dynamics similar to those considered by this paper [13], and examines implementations that function across different communication network topologies [24]. DMPC-based methods have also been used for trajectory generation for multi-agent UAV systems as a means of reducing computational time [25]. Outside the scope of UAVs, DMPC has also been used for the control of systems with *coupled dynamics* [26], [27], an important distinction between the collaborative payload transport algorithm proposed here and the bulk of prior UAV DMPC literature.

[1] Jad Wehbeh and Shatil Rahman are undergraduates at the Department of Mechanical Engineering, McGill University, Montreal, QC H3A 2K7, Canada jad.wehbeh|shatil.rahman@mail.mcgill.ca

[2] Inna Sharf is a Professor at the Department of Mechanical Engineering, McGill University, Montreal, QC H3A 2K7, Canada inna.sharf@mcgill.ca

978-1-7281-6211-9/20/$31.00 ©2020 IEEE

C. In this Paper

The key contribution of this paper is a novel distributed MPC formulation which allows for the minimization of a nonlinear cost function by coupled agents through state-specific linearization. The controller also relies on sharing predicted control inputs instead of state information to account for the coupled nature of the system. The paper is organized as follows. In Section II, we define the mathematical model for the kinematics and dynamics of the multi-UAV cooperative transport problem based on the work in [9] with some modifications. In Section III, a centralized MPC formulation is presented, followed by the novel distributed MPC algorithm in Section IV. Section V then provides a comparison of the performance of the centralized and distributed algorithms presented on an example maneuver.

II. MODELING

The dynamics model of the system presented here is motivated by the model developed in [9] and adheres to the same basic assumptions. In particular, we consider a system of N quadrotor UAVs that are carrying a single payload, the latter assumed to be a rigid body. The UAVs are connected to the payload via massless, rigid links, each attached at its ends to the quadrotor and payload via passive two-degree-of-freedom rotational or spherical joints. These effectively decouple the attitude motion of the vehicles from that of the payload, allowing for easier control of the system. Differently from the model developed in [9], we formulate the motion equations for the payload, as well as the links, by resolving them in the payload body-fixed frame (see Figure 1).

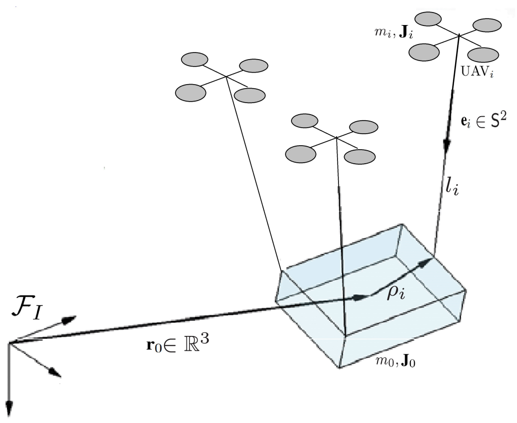

Fig. 1. Schematic of UAVs-Payload System

A. System Description

Throughout this paper, the variables related to the payload are denoted by the subscript 0, and the variables for the i-th quadrotor are denoted by the subscript i.

An inertial reference frame $\mathscr{F}_I$ and body-fixed frames $\mathscr{F}_{Bj}$ with basis vectors $\left\{\underset{\rightarrow}{\mathbf{b}}_{j1}, \underset{\rightarrow}{\mathbf{b}}_{j2}, \underset{\rightarrow}{\mathbf{b}}_{j3}\right\}$ for $0 \leq j \leq N$ are chosen. For the inertial frame, the third axis points downward along the direction of gravity. The origin of the j-th body-fixed frame is located at the center of mass of the payload for $j = 0$ and at the center of mass of the quadrotor for $1 \leq j \leq N$. The third body-fixed axis $\underset{\rightarrow}{\mathbf{b}}_{j3}$ is normal to the plane defined by the centers of the rotors, and it points downward.

As indicated in Figure 1, the location of the mass center of the payload in the inertial frame is denoted by $\mathbf{r}_0 \in \mathbb{R}^3$, and its attitude is described by $\mathbf{R}_0 \in \mathrm{SO}(3)$, the rotation matrix which transforms a vector expressed in the 0-th frame to the inertial frame. Let $\boldsymbol{\rho}_i \in \mathbb{R}^3$ be the point on the payload where the i-th link is attached, expressed in the payload-fixed frame, with the other end attached to the mass center of the i-th quadrotor. The unit vector from the center of mass of the i-th vehicle in the direction of the link, expressed in the payload frame, is denoted by $\mathbf{e}_i \in S^2$, where $S^2 = \left\{\mathbf{e} \in \mathbb{R}^3 | \|\mathbf{e}\| = 1\right\}$, and the fixed length of the i-th link is denoted by $l_i \in \mathbb{R}$.

Let $\mathbf{r}_i \in \mathbb{R}^3$ be the location of the mass center of the i-th quadrotor with respect to the inertial frame. Then, since the links are assumed to be rigid, the position kinematics follow directly as

$$\mathbf{r}_i = \mathbf{r}_0 + \mathbf{R}_0 \left(\boldsymbol{\rho}_i - l_i \mathbf{e}_i\right). \tag{1}$$

The attitude of the i-th quadrotor is defined by $\mathbf{R}_i \in \mathrm{SO}(3)$. In this payload-centric description of the system, the independent generalized coordinates are the position and orientation of the payload, the directions to the centers of mass of the vehicles, as defined by the $\mathbf{e}_i$ components in the payload frame and the orientation of the vehicles themselves. The configuration manifold of this system is therefore $\mathbb{R}^3 \times \mathrm{SO}(3) \times \left(\mathrm{S}^2 \times \mathrm{SO}(3)\right)^N$ and the system has $6 + 5N$ degrees of freedom.

The mass and the inertia matrix of the payload are denoted by $m_0 \in \mathbb{R}$ and $\mathbf{J}_0 \in \mathbb{R}^{3\times 3}$, respectively. The mass and the inertia matrix of the i-th quadrotor are denoted by $m_i \in \mathbb{R}$ and $\mathbf{J}_i \in \mathbb{R}^{3\times 3}$, respectively. The i-th quadrotor can generate a thrust $\boldsymbol{f}_i = f_i \mathbf{R}_i \mathbf{b}_{i3} \in \mathbb{R}^3$ expressed in its inertial frame, where $f_i \in \mathbb{R}$ is the total thrust generated by vehicle i and $\mathbf{b}_{i3} = [0 \;\; 0 \;\; 1]^T \in \mathbb{R}^3$. The propellers also generate a moment $\boldsymbol{\tau}_i \in \mathbb{R}^3$ with respect to the vehicle's body-fixed frame. The control inputs of this system are thus $\{f_i, \boldsymbol{\tau}_i\}$ for $1 \leq i \leq N$.

B. Kinematics and Dynamics

The kinematics equations for the system are given by:

$$\dot{\mathbf{r}}_0 = \mathbf{R}_0 \mathbf{v}_0 \tag{2}$$

$$\dot{\mathbf{e}}_i = \left(\boldsymbol{\Omega}_i - \boldsymbol{\omega}_0\right)^{\times} \mathbf{e}_i \tag{3}$$

$$\dot{\mathbf{r}}_i = \dot{\mathbf{r}}_0 + \mathbf{R}_0 \left(\boldsymbol{\omega}_0^{\times} \boldsymbol{\rho}_i - l_i \boldsymbol{\Omega}_i^{\times} \mathbf{e}_i\right) \tag{4}$$

$$\dot{\mathbf{R}}_0 = \mathbf{R}_0 \boldsymbol{\omega}_0^{\times} \tag{5}$$

$$\dot{\mathbf{R}}_i = \mathbf{R}_i \boldsymbol{\omega}_i^{\times} \tag{6}$$

where $\mathbf{v}_0$ is the translational velocity of the payload expressed in $\mathscr{F}_I$, $\boldsymbol{\omega}_0$ is the angular velocity of the payload and $\boldsymbol{\Omega}_i$ is the angular velocity of the i-th link, satisfying $\mathbf{e}_i^T \cdot \boldsymbol{\Omega}_i = 0$. The aforementioned variables are expressed in the payload frame, while the velocities of the i-th quadrotor and the payload are expressed in the inertial frame; the $^{\times}$ operator denotes the skew-symmetric matrix operator.

The equations of motion for the system are comprised of the following 4 sets: translational equations of motion of the payload, rotational equations of motion of the payload, rotational equations of motion of the links and rotational equations of motion of the quadrotors. They can be derived by applying the Newton-Euler formulation to the payload and to the vehicles, or alternatively using Lagrange's Equations with quasi-coordinates, and take the form:

$$\begin{aligned} m_T\left(\dot{\mathbf{v}}_0+\boldsymbol{\omega}_0^\times\mathbf{v}_0\right)+\sum_{i=1}^{N} m_i\left(-\boldsymbol{\rho}_i^\times\dot{\boldsymbol{\omega}}_0+l_i\mathbf{e}_i^\times\dot{\boldsymbol{\Omega}}_i+l_i\mathbf{e}_i^\times\boldsymbol{\omega}_0^\times\boldsymbol{\Omega}_i\right) \\ +\sum_{i=1}^{N} m_i\left(\boldsymbol{\omega}_0^\times\right)^2\boldsymbol{\rho}_i+m_il_i\left\|\boldsymbol{\Omega}_i\right\|^2\mathbf{e}_i=m_Tg\mathbf{R}_0^T\mathbf{k}+\sum_{i=1}^{N}\mathbf{R}_0^T\boldsymbol{f}_i, \end{aligned} \tag{7}$$

$$\begin{aligned} \bar{\mathbf{J}}_0\dot{\boldsymbol{\omega}}_0+\boldsymbol{\omega}_0^\times\bar{\mathbf{J}}_0\boldsymbol{\omega}_0+\sum_{i=1}^{N} m_i\boldsymbol{\rho}_i^\times(\dot{\mathbf{v}}_0+\boldsymbol{\omega}_0^\times\mathbf{v}_0+l_i\mathbf{e}_i^\times\dot{\boldsymbol{\Omega}}_i \\ +l_i\mathbf{e}_i^\times\boldsymbol{\omega}_0^\times\boldsymbol{\Omega}_i+l_i\left\|\boldsymbol{\Omega}_i\right\|^2\mathbf{e}_i)=\sum_{i=1}^{N}\boldsymbol{\rho}_i^\times\mathbf{R}_0^T\left(\boldsymbol{f}_i+m_ig\mathbf{k}\right), \end{aligned} \tag{8}$$

$$\begin{aligned} m_il_i^2\left(\dot{\boldsymbol{\Omega}}_i+\boldsymbol{\omega}_0^\times\boldsymbol{\Omega}_i\right)-m_il_i\mathbf{e}_i^\times\left(\dot{\mathbf{v}}_0+\boldsymbol{\omega}_0^\times\mathbf{v}_0\right)+m_il_i\mathbf{e}_i^\times\boldsymbol{\rho}_i^\times\dot{\boldsymbol{\omega}}_0 \\ -m_il_i\mathbf{e}_i^\times\left(\boldsymbol{\omega}_0^\times\right)^2\boldsymbol{\rho}_i=-l_i\mathbf{e}_i^\times\mathbf{R}_0^T\left(\boldsymbol{f}_i+m_ig\mathbf{k}\right), \end{aligned} \tag{9}$$

$$\mathbf{J}_i\dot{\boldsymbol{\omega}}_i+\boldsymbol{\omega}_i^\times\mathbf{J}_i\boldsymbol{\omega}_i=\boldsymbol{\tau}_i, \tag{10}$$

where $m_T=m_0+\sum_{i=1}^N m_i$, $\bar{\mathbf{J}}_0=\mathbf{J}_0-\sum_{i=1}^N m_i\left(\boldsymbol{\rho}_i^\times\right)^2$ and $\mathbf{k}=[0\ 0\ 1]^T$. This dynamics model of the system is similar to that in [9], except for $\mathbf{e}_i$, $\boldsymbol{\Omega}_i$ and $\boldsymbol{\omega}_0$ which have been expressed in the payload frame, rather than in the inertial one. This leads to the elimination of rotation matrices in a number of terms, with additional terms appearing in the equations of motion of the payload and the links. It is also noted that the rotational dynamics of the quadrotors, shown in Equation (10), are completely decoupled from the rest of the system.

To rewrite the model in state-space form for use with the controller formulation presented in Section III, we define the state of the system to be

$$\mathbf{x}=[\mathbf{r}_0^T,\mathbf{v}_0^T,\boldsymbol{\Theta}_0^T,\boldsymbol{\omega}_0^T,\mathbf{e}_i^T,\boldsymbol{\Omega}_i^T,\boldsymbol{\Theta}_i^T,\boldsymbol{\omega}_i^T]^T,\ \ i=1,\ldots,N$$

where $\boldsymbol{\Theta}_0$ and $\boldsymbol{\Theta}_i$ represent the 3-2-1 Euler angle parametrizations of $\mathbf{R}_0$ and $\mathbf{R}_i$, respectively. Equations (2) - (6) are trivial to write in state-space form, as is (10). The coupling dynamics pertaining to the kinetic states of the payload and the links, $\mathbf{x}_{0q}=[\mathbf{v}_0^T,\boldsymbol{\omega}_0^T,\boldsymbol{\Omega}_i^T]^T$, given in (7)-(9) can be written compactly as

$$\dot{\mathbf{x}}_{0q}=\mathbf{M}_{0q}^{-1}\mathbf{f}_{0q}(\mathbf{x},\mathbf{u}), \tag{11}$$

where

$$\mathbf{M}_{0q}=$$
$$\left[\begin{array}{cc|ccc} m_T\mathbf{1}_3 & -\sum_{i=1}^N m_i\boldsymbol{\rho}_i^\times & m_1l_1\mathbf{e}_1^\times & \ldots & m_Nl_N\mathbf{e}_N^\times \\ \sum_{i=1}^N m_i\boldsymbol{\rho}_i^\times & \bar{\mathbf{J}}_0 & m_1l_1\boldsymbol{\rho}_1^\times\mathbf{e}_1^\times & \ldots & m_Nl_N\boldsymbol{\rho}_N^\times\mathbf{e}_N^\times \\ \hline -m_1l_1\mathbf{e}_1^\times & m_1l_1\mathbf{e}_1^\times\boldsymbol{\rho}_1^\times & m_1l_1^2\mathbf{1}_3 & \ldots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ -m_Nl_N\mathbf{e}_N^\times & m_Nl_N\mathbf{e}_N^\times\boldsymbol{\rho}_N^\times & \mathbf{0} & \ldots & m_Nl_N^2\mathbf{1}_3 \end{array}\right],$$

and

$$\mathbf{f}_{0q}(\mathbf{x},\mathbf{u})=$$
$$\left[\begin{array}{c} -m_T\boldsymbol{\omega}_0^\times\mathbf{v}_0-\sum_{i=1}^N\{m_i\left(\boldsymbol{\omega}_0^\times\right)^2\boldsymbol{\rho}_i+m_il_i\left\|\boldsymbol{\Omega}_i\right\|^2\mathbf{e}_i \\ +m_il_i\mathbf{e}_i^\times\boldsymbol{\omega}_0^\times\boldsymbol{\Omega}_i\}+m_Tg\mathbf{R}_0^T\mathbf{k}+\sum_{i=1}^N\mathbf{R}_0^T\boldsymbol{f}_i \\ \hline -\boldsymbol{\omega}_0^\times\bar{\mathbf{J}}_0\boldsymbol{\omega}_0-\sum_{i=1}^N m_i\{\boldsymbol{\rho}_i^\times\boldsymbol{\omega}_0^\times\mathbf{v}_0+l_i\boldsymbol{\rho}_i^\times\mathbf{e}_i^\times\boldsymbol{\omega}_0^\times\boldsymbol{\Omega}_i \\ +l_i\boldsymbol{\rho}_i^\times\left\|\boldsymbol{\Omega}_i\right\|^2\mathbf{e}_i\}+\sum_{i=1}^N\boldsymbol{\rho}_i^\times\mathbf{R}_0^T\left(\boldsymbol{f}_i+m_ig\mathbf{k}\right) \\ \hline m_1l_1\left\{\mathbf{e}_1^\times\left(\boldsymbol{\omega}_0^\times\right)^2\boldsymbol{\rho}_1-l_1\boldsymbol{\omega}_0^\times\boldsymbol{\Omega}_1+\mathbf{e}_1^\times\boldsymbol{\omega}_0^\times\mathbf{v}_0\right\} \\ -l_1\mathbf{e}_1^\times\mathbf{R}_0^T\left(\boldsymbol{f}_1+m_1g\mathbf{k}\right) \\ \hline \vdots \\ \hline m_Nl_N\left\{\mathbf{e}_N^\times\left(\boldsymbol{\omega}_0^\times\right)^2\boldsymbol{\rho}_N-l_N\boldsymbol{\omega}_0^\times\boldsymbol{\Omega}_N+\mathbf{e}_N^\times\boldsymbol{\omega}_0^\times\mathbf{v}_0\right\} \\ -l_N\mathbf{e}_N^\times\mathbf{R}_0^T\left(\boldsymbol{f}_N+m_Ng\mathbf{k}\right) \end{array}\right].$$

C. Model Linearization

With the view to formulate a linear MPC in Section III, we develop a linearized model of the full system. Thus, assume the standard state and output equations

$$\begin{aligned} \dot{\mathbf{x}}&=\mathbf{f}(\mathbf{x}(t),\mathbf{u}(t)) \\ \mathbf{y}&=\mathbf{h}(\mathbf{x}(t),\mathbf{u}(t)) \end{aligned} \tag{12}$$

where $\mathbf{x}\in\mathbb{R}^n$, $\mathbf{u}\in\mathbb{R}^m$ and $\mathbf{f}(\mathbf{x}(t),\mathbf{u}(t))$ are the state, inputs, and dynamics model respectively, as defined in Sections II-A and II-B. $\mathbf{y}\in\mathbb{R}^p$ is the desired system output and represents the cost function to be minimized.

The model can be linearized about an arbitrary (non-equilibrium) state, input pair $(\mathbf{x}_0,\mathbf{u}_0)\in\mathbb{R}^n\times\mathbb{R}^m$ for use with the controller presented in Section III. The system is linearized by taking its first-order Taylor approximation about $(\mathbf{x}_0,\mathbf{u}_0)$ to obtain the relations

$$\begin{aligned} \dot{\mathbf{x}}&\approx\mathbf{f}(\mathbf{x}_0,\mathbf{u}_0)+\mathbf{A}_c\delta\mathbf{x}+\mathbf{B}_c\delta\mathbf{u} \\ \mathbf{y}&\approx\mathbf{h}(\mathbf{x}_0,\mathbf{u}_0)+\mathbf{C}\delta\mathbf{x}+\mathbf{D}\delta\mathbf{u}. \end{aligned} \tag{13}$$

where $\delta\mathbf{x}\in\mathbb{R}^n$ and $\delta\mathbf{u}\in\mathbb{R}^m$ are the respective deviations of $\mathbf{x}$, $\mathbf{u}$ from the linearization conditions, and $\mathbf{A}_c\in\mathbb{R}^{n\times n}$, $\mathbf{B}_c\in\mathbb{R}^{n\times m}$, $\mathbf{C}\in\mathbb{R}^{p\times n}$, and $\mathbf{D}\in\mathbb{R}^{p\times m}$ are the continuous-time linear system matrices.

III. CENTRALIZED MPC FORMULATION

The MPC algorithm presented in this section solves a linearly constrained quadratic program at each sampling time (iteration) to determine the approximate optimal control input. The formulation is based on the work in [28], extended to systems with nonlinear output equations and constraints, linearized about non-equilibrium points. The linearized model of the system in (13) is used to obtain expressions for the state and output as a function of the control input over the next N_p time steps. These are subsequently substituted into a target cost function which is minimized to obtain the optimal control over the prediction horizon. The first set of generated inputs is then executed before the process is repeated. Since the results of the optimization depend on the linearization, the dynamics, output, and constraints are relinearized and the cost is rebuilt at every iteration of the MPC.

A. Prediction Equations

For a given sampling interval $T_s > 0$, the discretized state dynamics and output equation equivalent to the linear system of (13) at time step k can be written as:

$$\begin{aligned}\delta\mathbf{x}_{k+1} &= \mathbf{f}(\mathbf{x}_0,\mathbf{u}_0)T_s + \mathbf{A}\delta\mathbf{x}_k + \mathbf{B}\delta\mathbf{u}_k\\ \mathbf{y}_k &= \mathbf{h}(\mathbf{x}_0,\mathbf{u}_0) + \mathbf{C}\delta\mathbf{x}_k + \mathbf{D}\delta\mathbf{u}_k.\end{aligned} \tag{14}$$

where $\delta\mathbf{x}_k = \mathbf{x}_k - \mathbf{x}_0$ is the difference between the state of the system at k and the linearization state, and $\delta\mathbf{u}_k = \mathbf{u}_k - \mathbf{u}_0$ is the change in control input from the linearization input.

The state prediction after N_p time steps is therefore

$$\delta\mathbf{x}_{k+N_p} = \mathbf{A}^{N_p}\delta\mathbf{x}_k + \sum_{j=1}^{N_p}\mathbf{A}^{j-1}[\mathbf{f}(\mathbf{x}_0,\mathbf{u}_0)T_s + \mathbf{B}\delta\mathbf{u}_{k+N_p-j}] \tag{15}$$

and the global prediction equations for the state and output are

$$\begin{aligned}\delta\mathbf{X}_k &= \mathbf{F} + \mathbf{G}\delta\mathbf{x}_k + \mathbf{H}\delta\mathbf{U}_k\\ \mathbf{Y}_k &= \mathbf{L} + \bar{\mathbf{C}}\delta\mathbf{X}_k + \bar{\mathbf{D}}\delta\mathbf{U}_k,\end{aligned} \tag{16}$$

where

$$\delta\mathbf{X}_k = \begin{bmatrix}\delta\mathbf{x}_k\\ \delta\mathbf{x}_{k+1}\\ \delta\mathbf{x}_{k+2}\\ \vdots\\ \delta\mathbf{x}_{k+N_p-1}\end{bmatrix}, \delta\mathbf{U}_k = \begin{bmatrix}\delta\mathbf{u}_k\\ \delta\mathbf{u}_{k+1}\\ \delta\mathbf{u}_{k+2}\\ \vdots\\ \delta\mathbf{u}_{k+N_p-1}\end{bmatrix}, \mathbf{G} = \begin{bmatrix}\mathbf{I}\\ \mathbf{A}\\ \mathbf{A}^2\\ \vdots\\ \mathbf{A}^{N_p-1}\end{bmatrix},$$

$$\mathbf{Y}_k = \begin{bmatrix}\mathbf{y}_k\\ \mathbf{y}_{k+1}\\ \mathbf{y}_{k+2}\\ \vdots\\ \mathbf{y}_{k+N_p-1}\end{bmatrix}, \mathbf{F} = \begin{bmatrix}\mathbf{0}\\ \mathbf{f}(\mathbf{x}_0,\mathbf{u}_0)T_s\\ \sum_{j=1}^{2}\mathbf{A}^{j-1}\mathbf{f}(\mathbf{x}_0,\mathbf{u}_0)T_s\\ \vdots\\ \sum_{j=1}^{N_p-1}\mathbf{A}^{j-1}\mathbf{f}(\mathbf{x}_0,\mathbf{u}_0)T_s\end{bmatrix},$$

$$\mathbf{L} = \begin{bmatrix}\mathbf{h}(\mathbf{x}_0,\mathbf{u}_0)\\ \mathbf{h}(\mathbf{x}_0,\mathbf{u}_0)\\ \mathbf{h}(\mathbf{x}_0,\mathbf{u}_0)\\ \vdots\\ \mathbf{h}(\mathbf{x}_0,\mathbf{u}_0)\end{bmatrix}, \mathbf{H} = \begin{bmatrix}\mathbf{0} & & & & \\ \mathbf{B} & \mathbf{0} & & & \\ \mathbf{AB} & \mathbf{B} & \mathbf{0} & & \\ \vdots & \vdots & \ddots & \ddots & \\ \mathbf{A}^{N_p-2}\mathbf{B} & \mathbf{A}^{N_p-3}\mathbf{B} & \dots & \mathbf{B} & \mathbf{0}\end{bmatrix},$$

$$\bar{\mathbf{C}} = \mathrm{diag}(\mathbf{C},\mathbf{C},\dots,\mathbf{C}), \bar{\mathbf{D}} = \mathrm{diag}(\mathbf{D},\mathbf{D},\dots,\mathbf{D}).$$

Here, the $\mathbf{F}$ and $\mathbf{L}$ matrices represent the main deviation from the work of [28].

B. Cost Function

The cost function minimized by the MPC algorithm penalizes the cumulative difference between the output of the system and the desired output $\mathbf{y}_k^t$ with weighting matrix $\mathbf{Q} \geq \mathbf{0} \in \mathbb{R}^{p\times p}$, and the norm of the control input with weighting matrix $\mathbf{R} > \mathbf{0} \in \mathbb{R}^{m\times m}$. $\mathbf{Q}_f \geq \mathbf{0} \in \mathbb{R}^{p\times p}$ is the cost weighting matrix on the terminal output. For the rest of this paper, $\mathbf{D}$ is taken to be zero. The cost function is therefore

$$\begin{aligned}\mathbf{J}(\delta\mathbf{X}_k,\delta\mathbf{U}_k) = {}& [\mathbf{L} + \bar{\mathbf{C}}\delta\mathbf{X}_k - \mathbf{Y}_k^t]^T\bar{\mathbf{Q}}[\mathbf{L} + \bar{\mathbf{C}}\delta\mathbf{X}_k - \mathbf{Y}_k^t]\\ &+ [\mathbf{h}(\mathbf{x}_0,\mathbf{u}_0) + \mathbf{C}(\delta\mathbf{x}_{k+N_p}) - \mathbf{y}_{k+N_p}^t]^T\mathbf{Q}_f\\ &[\mathbf{h}(\mathbf{x}_0,\mathbf{u}_0) + \mathbf{C}(\delta\mathbf{x}_{k+N_p}) - \mathbf{y}_{k+N_p}^t]\\ &+ (\mathbf{U}_0 + \delta\mathbf{U}_k)^T\bar{\mathbf{R}}(\mathbf{U}_0 + \delta\mathbf{U}_k),\end{aligned} \tag{17}$$

where

$$\mathbf{Y}_k^t = \begin{bmatrix}\mathbf{y}_k^t & \mathbf{y}_{k+1}^t & y_{k+2}^t & \cdots & y_{k+N_p-1}^t\end{bmatrix}^T,$$

$$\mathbf{U}_0 = \begin{bmatrix}\mathbf{u}_0 & \mathbf{u}_0 & \mathbf{u}_0 & \dots & \mathbf{u}_0\end{bmatrix}^T,$$

$$\bar{\mathbf{Q}} = \mathrm{diag}(\mathbf{Q},\mathbf{Q},\dots,\mathbf{Q}), \bar{\mathbf{R}} = \mathrm{diag}(\mathbf{R},\mathbf{R},\dots,\mathbf{R}).$$

Equation (16) can then be substituted into (17) to obtain a quadratic function in $\delta\mathbf{U}_k$ which serves as the target for the MPC.

C. Actuator Constraints

Consider the nonlinear input and state constraints

$$\begin{aligned}\mathbf{f_u}(\mathbf{U_k}) &\leq \mathbf{0}\\ \mathbf{g_u}(\mathbf{U_k}) &= \mathbf{0},\end{aligned} \tag{18}$$

$$\begin{aligned}\mathbf{f_x}(\mathbf{X_k}) &\leq \mathbf{0}\\ \mathbf{g_x}(\mathbf{X_k}) &= \mathbf{0}.\end{aligned} \tag{19}$$

Expanding $\mathbf{X}_k$ in Equation (17) into $\mathbf{X}_0 + \delta\mathbf{X}_k$ and substituting in (16) then allows the cost to be rewritten as a quadratic function in $\delta\mathbf{U}_k$ which serves as the target for the MPC.

Expanding $\mathbf{X}_k$ in Equation (19) into $\mathbf{X}_0 + \delta\mathbf{X}_k$ and substituting in (16) allows the state constraints to be reformulated as a function of the control inputs, and the result can be linearized about the vectors of repeated initial conditions $(\mathbf{X}_0,\mathbf{U}_0)$ to obtain the linear constraints

$$\begin{bmatrix}\nabla\mathbf{f_u}(\mathbf{U}_0)\\ \nabla\mathbf{f'_x}(\mathbf{U}_0)\end{bmatrix}\delta\mathbf{U}_k \leq \begin{bmatrix}-\mathbf{f_u}(\mathbf{U}_0)\\ -\mathbf{f_x}(\mathbf{X}_0)\end{bmatrix} \tag{20}$$

$$\begin{bmatrix}\nabla\mathbf{g_u}(\mathbf{U}_0)\\ \nabla\mathbf{g'_x}(\mathbf{U}_0)\end{bmatrix}\delta\mathbf{U}_k = \begin{bmatrix}-\mathbf{g_u}(\mathbf{U}_0)\\ -\mathbf{g_x}(\mathbf{X}_0)\end{bmatrix}, \tag{21}$$

where $\nabla\mathbf{f_u}(\mathbf{U}_0)$ is the gradient of $\mathbf{f_u}$ evaluated at $\mathbf{U}_0$, $\mathbf{f'_x}(\mathbf{U_k}) = \mathbf{f_x}(\mathbf{X_k})$, and

$$\mathbf{X}_0 = \begin{bmatrix}\mathbf{x}_0 & \mathbf{x}_0 & \mathbf{x}_0 & \dots & \mathbf{x}_0\end{bmatrix}^T.$$

IV. DISTRIBUTED MPC FORMULATION

The MPC problem described in Section III can be broken down and solved approximately in a distributed manner to decrease computational cost. Instead of solving for the optimal control of the N UAVs centrally, each UAV can compute its own optimal input based on an assumption of the other UAVs' behaviour.

In this section, $\mathbf{u}_{0,i}$, $\delta\mathbf{u}_{k,i}$, $\mathbf{U}_{0,i}$, and $\delta\mathbf{U}_{k,i}$ will be used to refer to the respective parts of $\mathbf{u}_0$, $\delta\mathbf{u}_k$, $\mathbf{U}_0$ and $\delta\mathbf{U}_k$ that correspond to the control inputs of UAV i. Additionally, $\delta\mathbf{x}_{k,i}$,

$\mathbf{y}_{k,i}$, $\delta\mathbf{X}_{k,i}$, and $\mathbf{Y}_{k,i}$ will denote the estimates of $\delta\mathbf{x}_k$, $\mathbf{y}_k$, $\delta\mathbf{X}_k$, and $\mathbf{Y}_k$ held by UAV i, and $\mathbf{B}_i$ will describe the matrix consisting of the columns of $\mathbf{B}$ corresponding to control inputs $\mathbf{u}_i$. Finally, the over-barred quantities, $\delta\bar{\mathbf{u}}_{k,i}$ and $\delta\bar{\mathbf{U}}_{k,i}$ will also be used to describe the control inputs generated by the other UAVs, where

$$\delta\bar{\mathbf{u}}_{k,i} = \begin{bmatrix} \delta\mathbf{u}_{k,1} \\ \vdots \\ \delta\mathbf{u}_{k,i-1} \\ \delta\mathbf{u}_{k,i+1} \\ \vdots \\ \delta\mathbf{u}_{k,N} \end{bmatrix}, \; \delta\bar{\mathbf{U}}_{k,i} = \begin{bmatrix} \delta\bar{\mathbf{u}}_{k,i} \\ \delta\bar{\mathbf{u}}_{k+1,i} \\ \delta\bar{\mathbf{u}}_{k+2,i} \\ \vdots \\ \delta\bar{\mathbf{u}}_{k+N_p-1,i} \end{bmatrix}$$

A. Communication

The distributed MPC described in this paper assumes a complete network graph, in which each UAV is capable of transmitting the required information to every other UAV. Each UAV is also assumed to have perfect information about the behaviour of the payload.

At every MPC iteration at t_k, each UAV receives the control generated by the other UAVs at the previous time step $\delta\bar{\mathbf{U}}_{k-1,i}$, and returns its own optimal control input sequence $\delta\mathbf{U}_{k,i}$. This sharing of control input predictions instead of state predictions distinguishes the algorithm presented from the majority of DMPC formulations, allowing it to be used for agents with coupled dynamics. State information is also exchanged to maintain an accurate estimate of the state across the different platforms, as shown in Figure 2.

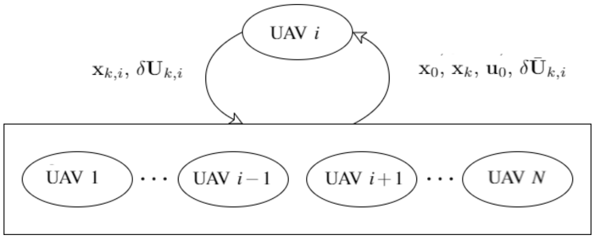
UAV i
$\mathbf{x}_{k,i}$, $\delta\mathbf{U}_{k,i}$
$\mathbf{x}_0$, $\mathbf{x}_k$, $\mathbf{u}_0$, $\delta\bar{\mathbf{U}}_{k,i}$
UAV 1
$\cdots$
UAV $i-1$
UAV $i+1$
$\cdots$
UAV N

Fig. 2. UAV Communication

Since the control input $\delta\bar{\mathbf{U}}_{k-1,i}$ received by UAV i is one time step behind the current state of the system, its final control input is held constant to generate a new input sequence $\delta\bar{\mathbf{U}}'_{k,i}$ for use in the optimization, where

$$\delta\bar{\mathbf{U}}'_{k,i} = \begin{bmatrix} \delta\bar{\mathbf{u}}_{k,i} \\ \delta\bar{\mathbf{u}}_{k+1,i} \\ \vdots \\ \delta\bar{\mathbf{u}}_{k+N_p-2,i} \\ \delta\bar{\mathbf{u}}_{k+N_p-2,i} \end{bmatrix}.$$

B. Input Optimization

The discretized state and output equations can be broken down to isolate $\delta\mathbf{u}_{k,i}$

$$\begin{aligned} \delta\mathbf{x}_{k+1,i} &= \mathbf{f}(\mathbf{x}_0,\mathbf{u}_0)T_s + \mathbf{A}\delta\mathbf{x}_{k,i} + \mathbf{B}_i\delta\mathbf{u}_{k,i} + \bar{\mathbf{B}}_i\delta\bar{\mathbf{u}}_{k,i} \\ \mathbf{y}_{k,i} &= \mathbf{h}(\mathbf{x}_0,\mathbf{u}_0) + \mathbf{C}\delta\mathbf{x}_{k,i}, \end{aligned} \quad (22)$$

and the prediction equations can be reformulated as

$$\begin{aligned} \delta\mathbf{x}_{k+N_p,i} = \mathbf{A}^{N_p}\delta\mathbf{x}_{k,i} + \sum_{j=1}^{N_p}\mathbf{A}^{j-1}[&\mathbf{f}(\mathbf{x}_0,\mathbf{u}_0)T_s \\ &+ \mathbf{B}\delta\mathbf{u}_{k+N_p-j,i} + \bar{\mathbf{B}}\delta\bar{\mathbf{u}}_{k+N_p-j,i}], \end{aligned} \quad (23)$$

$$\begin{aligned} \delta\mathbf{X}_{k,i} &= \mathbf{F} + \mathbf{G}\delta\mathbf{x}_k + \mathbf{H}_i\delta\mathbf{U}_{k,i} + \bar{\mathbf{H}}_i\delta\bar{\mathbf{U}}'_{k,i} \\ \mathbf{Y}_{k,i} &= \mathbf{L} + \bar{\mathbf{C}}\delta\mathbf{x}_{k,i}, \end{aligned} \quad (24)$$

where

$$\bar{\mathbf{B}}_i = \begin{bmatrix} \mathbf{B}_1 & \ldots & \mathbf{B}_{i-1} & \mathbf{B}_{i+1} & \ldots & \mathbf{B}_N \end{bmatrix},$$

$$\mathbf{H}_i = \begin{bmatrix} 0 & & & & \\ \mathbf{B}_i & 0 & & & \\ \mathbf{A}\mathbf{B}_i & \mathbf{B}_i & 0 & & \\ \vdots & \vdots & \ddots & \ddots & \\ \mathbf{A}^{N_p-2}\mathbf{B}_i & \mathbf{A}^{N_p-3}\mathbf{B}_i & \ldots & \mathbf{B}_i & 0 \end{bmatrix},$$

$$\bar{\mathbf{H}}_i = \begin{bmatrix} 0 & & & & \\ \bar{\mathbf{B}}_i & 0 & & & \\ \mathbf{A}\bar{\mathbf{B}}_i & \bar{\mathbf{B}}_i & 0 & & \\ \vdots & \vdots & \ddots & \ddots & \\ \mathbf{A}^{N_p-2}\bar{\mathbf{B}}_i & \mathbf{A}^{N_p-3}\bar{\mathbf{B}}_i & \ldots & \bar{\mathbf{B}}_i & 0 \end{bmatrix}.$$

Each UAV then attempts to minimize the same cost function as in the centralized case, but finds the optimal $\delta\mathbf{U}_{k,i}$ for a constant $\delta\bar{\mathbf{U}}'_{k-1,i}$. The cost is thus rewritten as

$$\begin{aligned} \mathbf{J}(\delta\mathbf{X}_{k,i},\delta\mathbf{U}_{k,i}) = [\mathbf{L} + \bar{\mathbf{C}}\delta\mathbf{X}_{k,i} - \mathbf{Y}^t_k]^T\bar{\mathbf{Q}}[\mathbf{L} + \bar{\mathbf{C}}\delta\mathbf{X}_{k,i} - \mathbf{Y}^t_k] \\ + [\mathbf{h}(\mathbf{x}_0,\mathbf{u}_0) + \mathbf{C}(\delta\mathbf{x}_{k+N_p,i}) - \mathbf{y}^t_{k+N_p}]^T\mathbf{Q}_f \\ [\mathbf{h}(\mathbf{x}_0,\mathbf{u}_0) + \mathbf{C}(\delta\mathbf{x}_{k+N_p,i}) - \mathbf{y}^t_{k+N_p}] \\ + (\mathbf{U}_{0,i} + \delta\mathbf{U}_{k,i})^T\bar{\mathbf{R}}(\mathbf{U}_{0,i} + \delta\mathbf{U}_{k,i}). \end{aligned} \quad (25)$$

The constraints of (20) and (21) are also reformulated in terms of $\delta\mathbf{U}_{k,i}$ to obtain

$$\begin{bmatrix} \nabla\mathbf{f}_{\mathbf{u},i}(\mathbf{U}_0) \\ \nabla\mathbf{f}'_{\mathbf{x},i}(\mathbf{U}_0) \end{bmatrix} \delta\mathbf{U}_{k,i} \leq \begin{bmatrix} -\mathbf{f}_{\mathbf{u}}(\mathbf{U}_0) - \nabla\bar{\mathbf{f}}_{\mathbf{u},i}(\mathbf{U}_0)\delta\bar{\mathbf{U}}'_{k-1,i} \\ -\mathbf{f}_{\mathbf{x}}(\mathbf{X}_0) - \nabla\bar{\mathbf{f}}'_{\mathbf{x},i}(\mathbf{U}_0)\delta\bar{\mathbf{U}}'_{k-1,i} \end{bmatrix}, \quad (26)$$

$$\begin{bmatrix} \nabla\mathbf{g}_{\mathbf{u},i}(\mathbf{U}_0) \\ \nabla\mathbf{g}'_{\mathbf{x},i}(\mathbf{U}_0) \end{bmatrix} \delta\mathbf{U}_{k,i} = \begin{bmatrix} -\mathbf{g}_{\mathbf{u}}(\mathbf{U}_0) - \nabla\bar{\mathbf{g}}_{\mathbf{u},i}(\mathbf{U}_0)\delta\bar{\mathbf{U}}'_{k-1,i} \\ -\mathbf{g}_{\mathbf{x}}(\mathbf{X}_0) - \nabla\bar{\mathbf{g}}'_{\mathbf{x},i}(\mathbf{U}_0)\delta\bar{\mathbf{U}}'_{k-1,i} \end{bmatrix}, \quad (27)$$

where $\nabla\mathbf{f}_{\mathbf{u},i}(\mathbf{U}_0)$ represents the columns of $\nabla\mathbf{f}_{\mathbf{u}}(\mathbf{U}_0)$ corresponding to control inputs $\delta\mathbf{U}_{k,i}$,

$$\nabla\bar{\mathbf{f}}_{\mathbf{u},i}(\mathbf{U}_0) = \begin{bmatrix} \nabla\mathbf{f}_{\mathbf{u},1}(\mathbf{U}_0) \\ \vdots \\ \nabla\mathbf{f}_{\mathbf{u},i-1}(\mathbf{U}_0) \\ \nabla\mathbf{f}_{\mathbf{u},i+1}(\mathbf{U}_0) \\ \vdots \\ \nabla\mathbf{f}_{\mathbf{u},N}(\mathbf{U}_0) \end{bmatrix}^T, \quad (28)$$

and $\nabla\mathbf{f}'_{\mathbf{x},i}(\mathbf{U}_0)$, $\nabla\bar{\mathbf{f}}'_{\mathbf{x},i}(\mathbf{U}_0)$, $\mathbf{g}_{\mathbf{u},i}(\mathbf{U}_0)$, $\nabla\bar{\mathbf{g}}_{\mathbf{u},i}(\mathbf{U}_0)$, $\nabla\mathbf{g}'_{\mathbf{x},i}(\mathbf{U}_0)$, and $\nabla\bar{\mathbf{g}}'_{\mathbf{x},i}(\mathbf{U}_0)$ are similarly defined.

V. SIMULATION RESULTS

Results are presented for a MATLAB simulation of the controllers, using the dynamics of the system defined in Section II-B. The performance of the distributed MPC controller is compared to that of the centralized controller under two sets of conditions. First, the distributed controller is evaluated at the same frequency as the centralized MPC. Then, the frequency of the distributed MPC is scaled to require the same computation time as the centralized controller.

A. System Definition and Maneuver

The system modeled to showcase the performance of the distributed controller is that of four identical UAVs connected to a single payload. UAVs begin in a square formation centered at and above the payload, which lies 10 m above the origin of the inertial reference frame. An N_p of 20 is chosen alongside a centralized T_s of 0.05 s based on experimentation. The characteristics of the system used in simulation are shown in Table I.

TABLE I
PARAMETERS OF UAVS AND PAYLOAD

	m (kg)	J_{xx} (kg.m^2)	J_{yy} (kg.m^2)	J_{zz} (kg.m^2)
Payload	3	0.556	0.556	0.556
UAVs	1.5	0.029	0.029	0.055
	$\boldsymbol{\rho}_i$ (m)			l_i (m)
UAV1	[0.25, 0.25, -0.125]			3.2
UAV2	[0.25, -0.25, -0.125]			3.2
UAV3	[-0.25, -0.25, -0.125]			3.2
UAV4	[-0.25, 0.25, -0.125]			3.2

The same maneuver is performed for the different controllers, with the UAVs initially hovering before climbing 2.5 m, translating 2.5 m in the positive x direction, descending 2.5 m, and returning to hover. A time frame of 5 seconds is allotted to each phase of the maneuver.

The cost function y is constructed to penalize the UAV position deviation from the specified path as well as payload attitude deviation from the target of zero yaw, pitch and roll. A slight cost is also placed on nonzero UAV yaw rates so as to ensure the strict convexity of the optimization problem. We choose to specify targets for both the UAVs and payload to demonstrate the algorithm's ability to simultaneously handle formation keeping and stabilization.

Maximum and minimum control input values are specified, as well as control input rate limits. Collisions between UAVs are prevented by adding a minimum position distance as a state inequality constraint.

B. Centralized MPC Results

As can be seen from Figures 3 and 4, the centralized controller successfully performs the maneuver prescribed. Payload oscillations are limited to within amplitudes of 2.5 degrees and the cost never exceeds a maximum of 3 after the formation stabilizes about the initial hover.

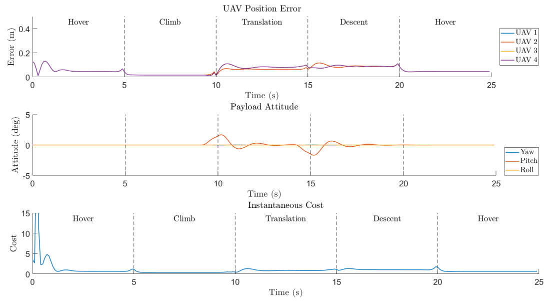

Fig. 3. Centralized Controller Performance

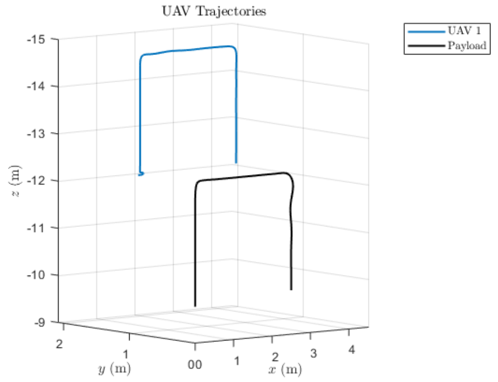

Fig. 4. Centralized Controller Trajectories

C. Distributed MPC Results

In the case of the distributed MPC controller, the results shown in Figure 5 are first generated for the controller operating under the same conditions as the centralized one. The trajectories for the UAVs and payload are nearly identical to those shown in Figure 4 and are not included. As expected, the controller performs slightly worse than the centralized implementation with larger tracking error and oscillation amplitudes, as well as a greater cost.

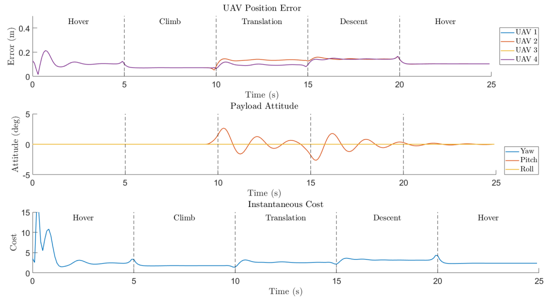

Fig. 5. Decentralized Controller Performance - Centralized Rate

The sampling time T_s is then scaled to achieve the same run time for the centralized and distributed controllers. The plots obtained are shown in Figure 6, with trajectories similar to those shown in Figure 4. Increasing the rate significantly ameliorates the performance of the algorithm, improving cost performance slightly compared to the centralized algorithm.

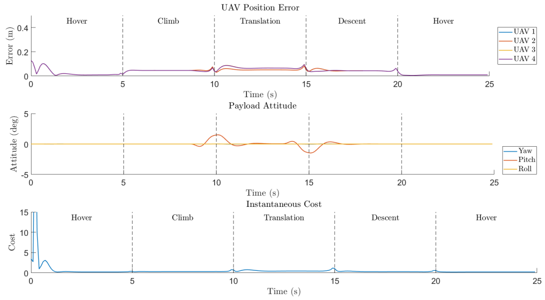

Fig. 6. Distributed Controller Performance - Scaled Rate

VI. CONCLUSIONS

A model for a system of quadrotor UAVs connected to a payload was presented, and an MPC controller was proposed based on the model's linearization. The controller was then generalized to run in a distributed manner, and simulation results were presented for the various controllers. Both the centralized and decentralized algorithms executed the maneuver successfully. The decentralized controller performed worse than the centralized one when run at the same frequency, but outperformed it marginally when the frequencies were scaled to require the same computation time across both algorithms.

ACKNOWLEDGMENT

This work was supported by the National Sciences and Engineering Research Council (NSERC) Canadian Robotics Network (NCRN) and Summer Undergraduate Research in Engineering (SURE) programs.

REFERENCES

[1] E. Ackerman and M. Koziol, "The blood is here: Zipline's medical delivery drones are changing the game in rwanda," *IEEE Spectrum*, vol. 56, no. 5, pp. 24–31, 2019.

[2] F. A. Goodarzi, D. Lee, and T. Lee, "Geometric control of a quadrotor UAV transporting a payload connected via flexible cable," *International Journal of Control, Automation and Systems*, vol. 13, no. 6, pp. 1486–1498, 2015.

[3] I. Palunko, R. Fierro, and P. Cruz, "Trajectory generation for swing-free maneuvers of a quadrotor with suspended payload: A dynamic programming approach," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 2691–2697.

[4] S. Tang, V. Wüest, and V. Kumar, "Aggressive flight with suspended payloads using vision-based control," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1152–1159, 2018.

[5] K. Yi, Y. He, J. Han, L. Yang, and F. Gu, "A review on control methods for multi-lift rotorcraft systems," in *2018 IEEE 8th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*. IEEE, 2018, pp. 913–918.

[6] K. Klausen, T. I. Fossen, and T. A. Johansen, "Suspended load motion control using multicopters," in *22nd Mediterranean Conference on Control and Automation*. IEEE, 2014, pp. 1371–1376.

[7] I. H. B. Pizetta, A. S. Brandão, and M. Sarcinelli-Filho, "Cooperative quadrotors carrying a suspended load," in *2016 international conference on unmanned aircraft systems (ICUAS)*. IEEE, 2016, pp. 1049–1055.

[8] K. Sreenath and V. Kumar, "Dynamics, control and planning for cooperative manipulation of payloads suspended by cables from multiple quadrotor robots," *rn*, vol. 1, no. r2, p. r3, 2013.

[9] T. Lee, "Geometric control of quadrotor UAVs transporting a cable-suspended rigid body," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 1, pp. 255–264, 2017.

[10] K. Klausen, C. Meissen, T. I. Fossen, M. Arcak, and T. A. Johansen, "Cooperative control for multirotors transporting an unknown suspended load under environmental disturbances," *IEEE Transactions on Control Systems Technology*, 2018.

[11] B. Shirani, M. Najafi, and I. Izadi, "Cooperative load transportation using multiple UAVs," *Aerospace Science and Technology*, vol. 84, pp. 158–169, 2019.

[12] I. H. B. Pizetta, A. S. Brandão, and M. Sarcinelli-Filho, "Avoiding obstacles in cooperative load transportation," *ISA transactions*, vol. 91, pp. 253–261, 2019.

[13] G. Tartaglione, E. D'Amato, M. Ariola, P. S. Rossi, and T. A. Johansen, "Model predictive control for a multi-body slung-load system," *Robotics and Autonomous Systems*, vol. 92, pp. 1–11, 2017.

[14] M. Gassner, T. Cieslewski, and D. Scaramuzza, "Dynamic collaboration without communication: Vision-based cable-suspended load transport with two quadrotors," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 5196–5202.

[15] L. Qian and H. H. Liu, "Path following control of multiple quadrotors carrying a rigid-body slung payload," in *AIAA Scitech 2019 Forum*, 2019, p. 1172.

[16] R. Scattolini, "Architectures for distributed and hierarchical model predictive control–a review," *Journal of process control*, vol. 19, no. 5, pp. 723–731, 2009.

[17] P. D. Christofides, R. Scattolini, D. M. de la Pena, and J. Liu, "Distributed model predictive control: A tutorial review and future research directions," *Computers & Chemical Engineering*, vol. 51, pp. 21–41, 2013.

[18] A. Richards and J. How, "Decentralized model predictive control of cooperating UAVs," in *2004 43rd IEEE Conference on Decision and Control (CDC)(IEEE Cat. No. 04CH37601)*, vol. 4. IEEE, 2004, pp. 4286–4291.

[19] A. Bemporad and C. Rocchi, "Decentralized linear time-varying model predictive control of a formation of unmanned aerial vehicles," in *2011 50th IEEE conference on decision and control and European control conference*. IEEE, 2011, pp. 7488–7493.

[20] G. Franzè, W. Lucia, and B. Rahami, "Distributed receding horizon control for rotating wings unmanned aerial vehicles: a time-varying topology strategy," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 3329–3334.

[21] S. S. Mansouri, G. Nikolakopoulos, and T. Gustafsson, "Distributed model predictive control for unmanned aerial vehicles," in *2015 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*. IEEE, 2015, pp. 152–161.

[22] T. T. Ribeiro, A. G. Conceição, I. Sa, and P. Corke, "Nonlinear model predictive formation control for quadcopters," *IFAC-PapersOnLine*, vol. 48, no. 19, pp. 39–44, 2015.

[23] Z. Cai, H. Zhou, J. Zhao, K. Wu, and Y. Wang, "Formation control of multiple unmanned aerial vehicles by event-triggered distributed model predictive control," *IEEE Access*, vol. 6, pp. 55 614–55 627, 2018.

[24] A. Grancharova, E. I. Grøtli, and T. A. Johansen, "Rotary-wing UAVs trajectory planning by distributed linear MPC with reconfigurable communication network topologies," *IFAC Proceedings Volumes*, vol. 46, no. 27, pp. 198–205, 2013.

[25] C. E. Luis and A. P. Schoellig, "Trajectory generation for multiagent point-to-point transitions via distributed model predictive control," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 375–382, 2019.

[26] A. Ibanez, P. Bidaud, and V. Padois, "A distributed model predictive control approach for robust postural stability of a humanoid robot," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 202–209.

[27] J. M. Maestre, D. Munoz De La Pena, and E. F. Camacho, "Distributed model predictive control based on a cooperative game," *Optimal Control Applications and Methods*, vol. 32, no. 2, pp. 153–176, 2011.

[28] P. Ru and K. Subbarao, "Nonlinear model predictive control for unmanned aerial vehicles," *Aerospace*, vol. 4, no. 2, p. 31, 2017.