Imperial College London

Department of Electrical and Electronic Engineering

Final Year Project Report 2019

Project Title:	**Multi-Agent Optimal Control for Cooperative Payload Transportation**
Student:	**Daniel Atticus Williams**
CID:	**01044837**
Course:	**EEE4-T**
Project Supervisor:	**Dr Eric C. Kerrigan**
Second Marker:	**Dr Adrià Junyent-Ferré**

Abstract

This project involves the application of model predictive control (MPC) to the problem of cooperative payload transportation (CPT). With recent improvements in the cost and reliability of unmanned aerial vehicle technology, an established body of research into robotic payload transportation has motivated the use of aerial agents for CPT. The primary goal of this work is to develop a novel MPC-based control architecture for aerial CPT schemes involving cable suspension of the load from multiple agents.

We propose a novel centralized control architecture by formulating a decreasing horizon optimal control problem (DHOCP) for optimizing the agents' trajectories. The DHOCP maintains compliance with secondary control objectives (maintaining connectivity between the agents and the payload, avoiding inter-agent collisions and spatial obstacles) by formulating such objectives as path constraints for each agent. We then present a staged implementation of the proposed architecture using the ICLOCS2 trajectory optimization package for simulations in MATLAB. Finally, we graphically derive mathematical models in Simscape for closed-loop simulations with Simulink, to verify the performance of the open-loop simulations and visualize the system's physical behavior.

Acknowledgements

I would like to express my deep gratitude to my academic supervisor Dr Eric Kerrigan and Ian McInerney for their guidance in this project. Their wisdom and patience throughout the year have been tremendous and invaluable. I must also thank Yuanbo Nie for his assistance in guiding me through the correct use of ICLOCS2.

The idea for this project was motivated by previous research conducted at the RISC Lab at KAUST in 2018, so I am grateful to Prof Jeff Shamma for his inspiration and for having introduced me to the world of cooperative robotics.

Ultimately I must thank my parents for their inexhaustable encouragement and support over these four years. You have always spurred me on to do my best.

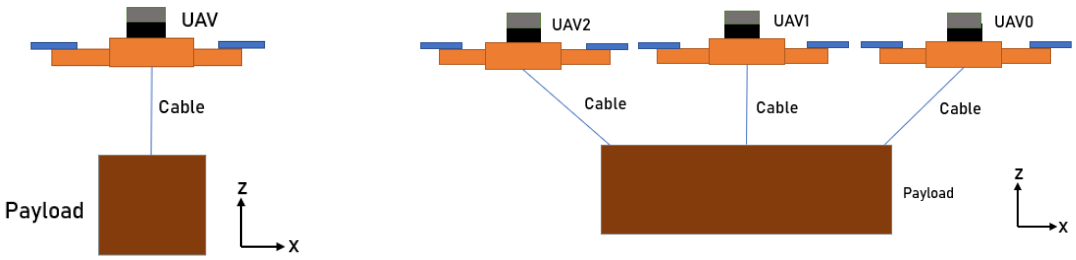
Contents

List of Figures

Chapter 1

Introduction

Transporting objects is a fundamental task in the field of robotics [1]. As a subset of the manipulation problem, payload transportation is conducted by mobile agents such as unmanned aerial vehicles (UAVs). Single-agent aerial payload transportation has been studied by numerous sources but such systems are inherently limited by the thrust capabilities of the agent's hardware as depicted in Figure 1.1a [2]. In recent years, increasing interest in aerial payload transportation from the commercial [3], industrial [4], agricultural [5], public health [6] and military [7] sectors has motivated the development of cooperative payload transportation (CPT) [8] (Figure 1.1b). CPT schemes exploit multiple agents' carrying capabilities to transport bulky and heavy objects. This allows for a wider set of possible transportation maneuvers at the cost of an increase in model complexity.



(a) Single-Agent System

(b) Multi-Agent System

Figure 1.1: Aerial Payload Transportation

1.1 Previous Work

The project is motivated by previous research conducted as part of a summer placement. In the earlier project we devised a decentralized control algorithm for CPT by two UAVs as depicted in Figure 1.2. The proposed control scheme employed one UAV as a 'leader' agent tracking a preset trajectory in one dimension and the second UAV as a 'follower' agent minimizing the deviations from a desired cable angle, both relying on the PID controller architecture.

While the control scheme yielded satisfactory performance in mission simulations, we noted three limitations:

- Controller parameters needed to be fine-tuned manually in an open-loop manner;
- Compliance with other objectives (such as payload swing oscillations and collision avoidance) could not be guaranteed;
- Optimality of the tracked trajectory was unknown.

The issues directly impacted the safety and efficiency of the CPT scheme implementation and could be addressed using optimal control theory. It has been this previous work which has motivated the application of model predictive control (MPC) to the CPT problem.

1.2 ICLOCS2

Software is required to explore how MPC affects the CPT problem. In this regard, of particular relevance is the ICLOCS2 trajectory optimization package for MATLAB [9][10]. The ICLOCS2 package allows users to systematically pose and solve an optimal control problem (OCP), in which a system evolves according to an analytically derived mathematical model and must comply with user-defined path constraints. The user may choose from a list of methods to transcribe the OCP into a nonlinear programming (NLP) problem. At present ICLOCS2 supports the NLP solvers *IPOPT*, *fmincon*, and *WORHP*. The user may also select from a range of discretization methods to accurately represent the system's states and inputs. Given the system's model, ICLOCS2 can automatically compute the derivative matrices for the model using finite difference methods, thereby simplifying the problem setup process for the user. A major advantage of ICLOCS2 is its interoperability with MATLAB, allowing for numerical simulations of the system in open-loop (when the optimal sequence of system inputs is calculated once only) and in closed-loop (when the optimal sequence of system inputs is periodically reoptimized over the mission duration). It is for these reasons we have decided to investigate the CPT problem using ICLOCS2 for trajectory optimization within an MPC framework.

1.3 Objectives

Our project goal is to design and simulate a novel optimal controller for a multi-agent CPT scheme, relying on MPC for real-time optimization of the agents' trajectories, obstacle avoidance, and prevention of inter-agent collisions. To this end, we have formulated the following objectives for the project:

- To implement the optimal controller using ICLOCS2 and numerically simulate its performance using Simulink [11].
- To create visual simulations of missions under the proposed CPT scheme using Simscape for Simulink [12].

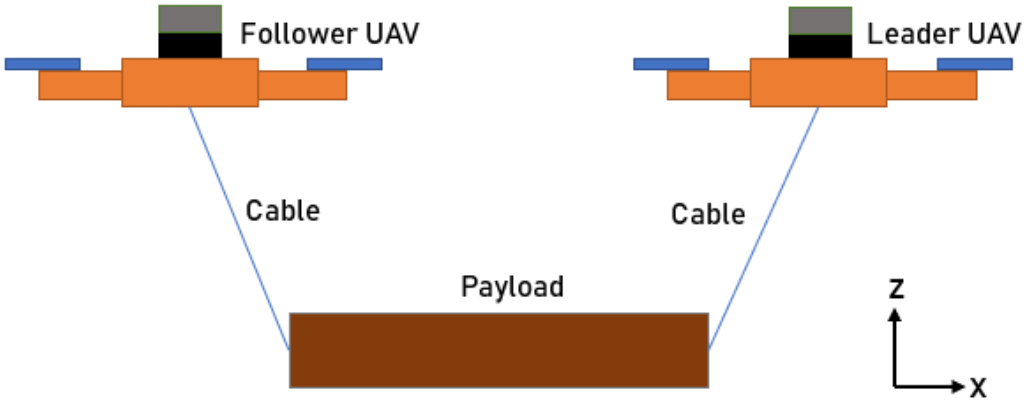


Figure 1.2: Cooperative Payload Transportation using Unmanned Aerial Vehicles

1.4 Deliverables

Pursuant to the stated project goals and objectives, we have sought to complete the following tasks:

1. Survey the academic literature concerning multi-agent CPT schemes using UAVs.
2. Create a set of design requirements for a centralized optimal multi-agent CPT scheme.
3. Satisfy the identified design requirements.
4. Analytically derive a mathematical model of the problem ('ADM model').
5. Create numerical simulations of the system with ICLOCS2 in closed-loop with the ADM model in Simulink.
6. Graphically derive a mathematical model ('GDM model') of the system in Simscape.
7. Create a visual simulation of system operation using ICLOCS2 in closed-loop with the GDM model in Simulink.

1.5 Project Outline

Chapter 2 introduces the fundamentals of cooperative payload transportation and summarizes the state of the art. The study of existing CPT schemes and their characteristics is used to extract performance requirements for an ideal CPT scheme. In Chapter 3 we formulate the problem statement and devise a centralized optimal control architecture that satisfies the identified design requirements. Chapter 4 details the stages of work towards implementing the proposed control architecture, progressing from single agent relocation towards multi-agent CPT. The procedures for testing the implementation and noteworthy results are discussed in Chapter 5. Finally, in Chapter 6 we conclude with a summary of the project's outcomes, evaluate the implementation's performance and compliance with the project objectives, and identify future work that could advance the project.

Chapter 2

Background

We introduce the fundamentals of cooperative payload transportation and summarize the state of the art in this chapter. We also consider existing CPT schemes and extract performance requirements for an ideal CPT scheme.

2.1 Cooperative Payload Transportation

CPT schemes may be classified according to their design (Figure 2.1) [13].

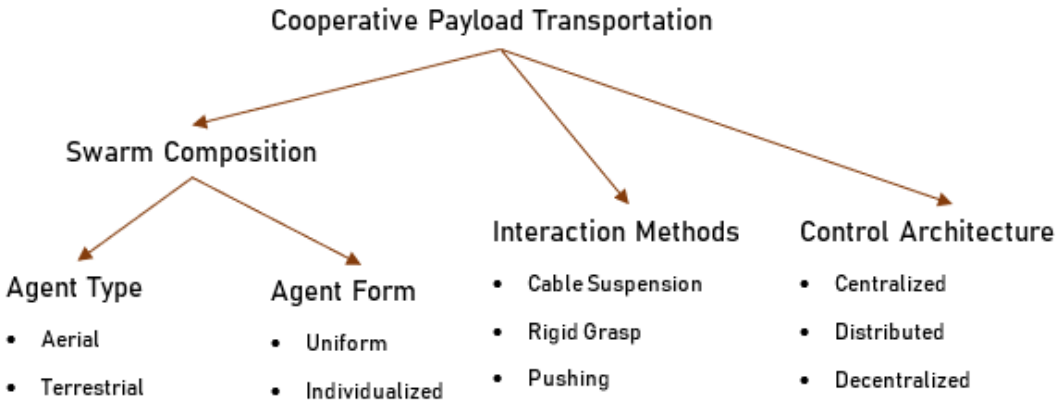
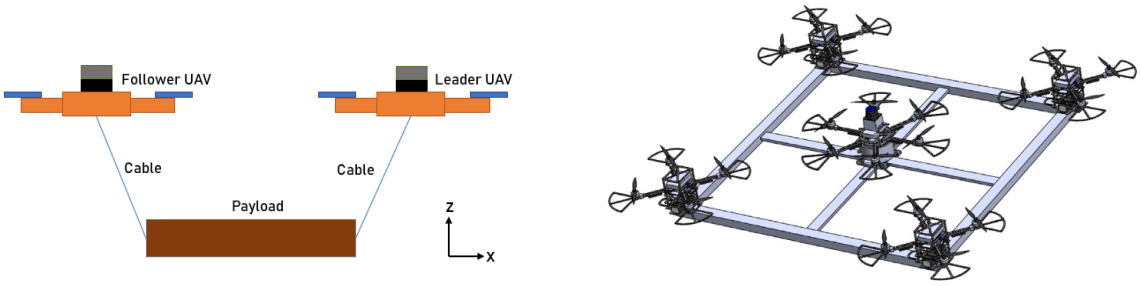
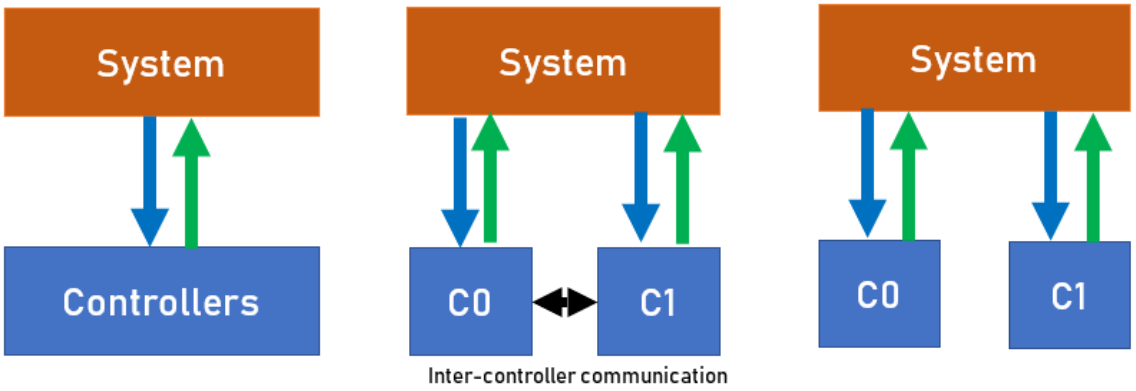


Figure 2.1: Taxonomy of CPT Schemes



(a) Cable-Suspended CPT (b) Rigid-Grasp CPT [15]

Figure 2.2: Interaction Methods for CPT Schemes



(a) Centralized Control (b) Distributed Control (c) Decentralized Control

Figure 2.3: Control Architectures

Swarm Composition: Choosing the agent type has an important impact on design. UAVs allow agile motion in three dimensions but may have shorter mission durations than terrestrial vehicles due to their limited onboard battery life. The composition of the formation must also be considered, i.e. whether each agent should have a uniform design or an individualized design (perhaps for task specializations, as seen in [14]).

Interaction Methods: Once a decision has been made about the formation structure, the next issue involves interacting with the payload (Figure 2.2a). In sling-load systems the payload is suspended by a cable that permits versatile tensile manipulation in three dimensions [8]. Systems that use rigid attachment to the agents' bodies allow direct inference of the payload's location but may not suit certain payload sizes and shapes (Figure 2.2b).

Control Architecture: How one chooses an appropriate control architecture is important [16]. Centralized control (Figure 2.3a) involves decisions made in one location that are communicated to all agents. Distributed control (Figure 2.3b) allows for each agent to make decisions with some communication with peers. Decentralized control (Figure 2.3c) requires each agent to make their own decisions without communicating with peers.

2.2 Requirements Capture

The literature has identified several important requirements for UAV-based CPT systems:

- Robustness in a range of environments [17];
- Minimal reliance on external infrastructure [17];
- To the extent possible, minimization or elimination of dependence on explicit communication between agents [18].

The following design requirements are also identified in [19]:

- Operability in unstructured indoor and outdoor environments;
- Robust flight capabilities;
- Autonomous, onboard decision making (this requires control algorithms with lower computational complexity [20]);
- Modular and flexible sensing and control;
- No dependence on external navigation aids.

Mission Objectives: Mission objectives occupy a hierarchy that is specific to UAV-based CPT schemes [21]:

1. Avoid obstacles (also featured in [2] and [20])
2. Secondary objectives:
   (a) Avoid collisions and excessive separation between agents
   (b) Evenly distribute payload weight between vehicles
3. Reduce oscillations caused by external disturbances such as wind (incompatible with fast and aggressive agent maneuvers [20])

Optimal control algorithms should also minimize agent thrust to conserve battery life.

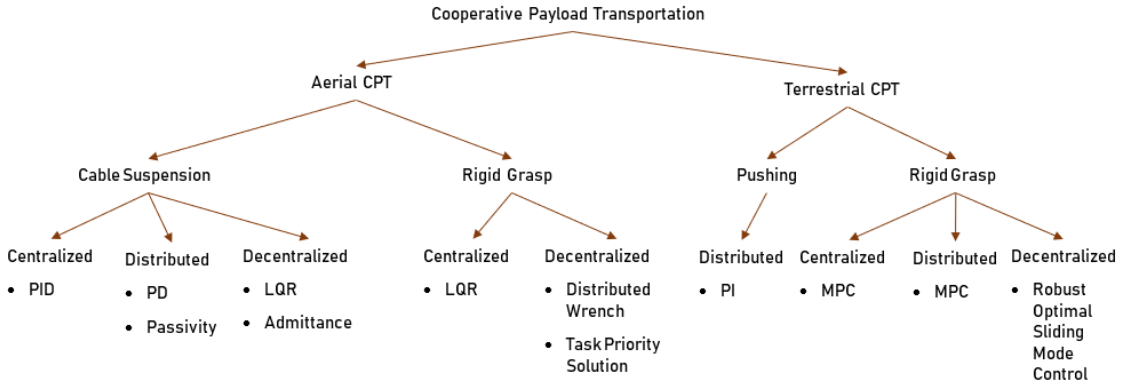


Figure 2.4: State of the Art for CPT Schemes

2.3 Existing CPT Schemes and Limitations

The literature proposes a selection of aerial- and terrestrial-based schemes representing various combinations of interaction methods, degrees of control centralization, and formation control algorithms (Figure 2.4).

Control Centralization: Current trends in UAV-based CPT research have seen a shift from centralized control to decentralized control. When centralized and distributed architectures rely on communication they have latencies, are susceptible to infrastructure failure and increase consumption of power and spectrum. Each problem may be avoided by eliminating explicit communication. However, fully decentralized architectures rely on alternative methods of mutual localization that are more computationally complex [22].

System Assumptions: All methods of implementation of CPT schemes make certain operational assumptions. Common environmental assumptions include the absence of obstacles and external disturbances such as wind. Most CPT schemes assume external infrastructure for agent localization ([23] is a notable exception). Few CPT schemes perform trajectory optimization in real-time ([24], [25]).

Aerial CPT: CPT schemes using cable suspension employ all three types of control architectures. Centralized architectures relying on PID controllers are proposed in [21], [26] and [27], while a nonlinear PD controller is proposed in [28]. Decentralized control underpins several recent proposals involving linear quadratic regulators (LQR) [18], [29], and admittance control [23], [30] (the latter uses MPC to track the reference trajectory generated by the admittance controller). Fewer proposals have implemented a distributed control architecture, using a PD controller [31] or a passivity-based approach [32]. In the CPT schemes which use rigid payload grasping, a centralized control architecture relying on LQR is presented in [15], while a distributed wrench controller is proposed in [33] and a distributed controller based on task priority solution features in [34].

Terrestrial CPT: A small number of proposed CPT schemes rely on terrestrial vehicles. A distributed controller for collaborative pushing movement using PI controllers is explored in [35]. Rigid-grasp systems appear in [36], [24] and [25]; the latter two rely on MPC techniques to solve the centralized CPT problem [24] and the distributed CPT problem [25].

Chapter 3

Analysis and Design

We formulate the problem statement in this chapter and devise a centralized optimal control architecture that satisfies the identified design requirements.

3.1 System Dynamics

We begin by defining an inertial coordinate system in $\mathbb{R}^3$ to describe the position of each agent in three dimensions. Unless otherwise stated, we adopt the 'East-North-Up' convention (Figure 3.1) and assume the gravitational constant $g = 9.81\,\text{m.s}^{-2}$.

Let there be N agents modeled by point masses m_{ag}; we assume $N = 3$ for this project unless otherwise stated. Consider the i^{th} agent located in the reference frame depicted in Figure 3.1, with position $(x_i, y_i z_i)$ and velocity $(\dot{x}_i, \dot{y}_i, \dot{z}_i)$. Define the instantaneous state vector for the i^{th} agent as $X_i = \begin{bmatrix} x_i^T & \dot{x}_i^T & y_i^T & \dot{y}_i^T & z_i^T & \dot{z}_i^T \end{bmatrix}^T$. We assume that each agent has an onboard flight controller that can maintain the agent's pose (position and orientation) in the presence of bounded external disturbances. We may therefore model each agent's dynamics as a set of double-integrators driven by three orthogonal inputs in the x-, y-, and z-axes. Define the input vector for the i^{th} agent as $U_i = \left[u_{xi}^T, u_{yi}^T, u_{zi}^T\right]^T$ [37].

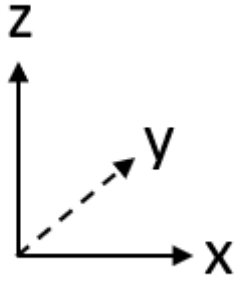
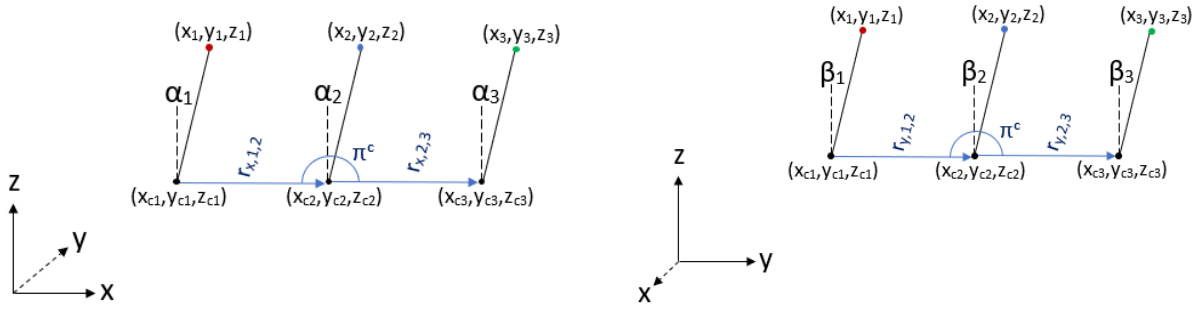


Figure 3.1: East-North-Up Coordinates for $\mathbb{R}^3$



(a) X-Z Plane

(b) Y-Z Plane

Figure 3.2: Payload Geometry

Next we assume that each agent is connected to a payload via inelastic cables of length $L > 0$ and negligible mass as depicted in Figure 3.2. We assume the payload is rigid and has a homogeneous composition. To simplify dynamical analysis, we approximate the payload using three identical points of mass $\frac{m_{pl}}{3}$ placed at the cable's attachment points (x_{c1}, y_{c1}, z_{c1}), (x_{c2}, y_{c2}, z_{c2}), and (x_{c3}, y_{c3}, z_{c3}), and collinearly distributed about the payload's center of mass. For the i^{th} agent, we define the angles α_i and β_i between its cable and the z-axis in the X-Z plane and Y-Z plane respectively. These are governed by a set of algebraic equations:

$$\alpha_1 = \text{atan}(\frac{x_{c1} - x_1}{z_{c1} - z_1}) \tag{3.1}$$

$$\beta_1 = \text{atan}(\frac{y_{c1} - y_1}{z_{c1} - z_1}) \tag{3.2}$$

$$\alpha_2 = \text{atan}(\frac{x_{c2} - x_2}{z_{c2} - z_2}) \tag{3.3}$$

$$\beta_2 = \text{atan}(\frac{y_{c2} - y_2}{z_{c2} - z_2}) \tag{3.4}$$

$$\alpha_3 = \text{atan}(\frac{x_{c3} - x_3}{z_{c3} - z_3}) \tag{3.5}$$

$$\beta_3 = \text{atan}(\frac{y_{c3} - y_3}{z_{c3} - z_3}) \tag{3.6}$$

Given the cable material's spring constant k, the cable tensions (F_1, F_2, F_3) resolved in the z-axis are governed by the following set of algebraic equations:

$$F_1 = \frac{k\sqrt{(x_{c1} - x_1)^2 + (y_{c1} - y_1)^2 + (z_{c1} - z_1)^2}}{\sqrt{1 + (\tan(\alpha_1))^2 + (\tan(\beta_1))^2}} \tag{3.7}$$

$$F_2 = \frac{k\sqrt{(x_{c2} - x_2)^2 + (y_{c2} - y_2)^2 + (z_{c2} - z_2)^2}}{\sqrt{1 + (\tan(\alpha_2))^2 + (\tan(\beta_2))^2}} \tag{3.8}$$

$$F_3 = \frac{k\sqrt{(x_{c3} - x_3)^2 + (y_{c3} - y_3)^2 + (z_{c3} - z_3)^2}}{\sqrt{1 + (\tan(\alpha_3))^2 + (\tan(\beta_3))^2}} \tag{3.9}$$

If we assume cable inelasticity we may approximate (F_1, F_2, F_3) with constant values.

By resolving the force components incident at the payload's center of mass, we can analytically derive the mathematical model for the payload's translational dynamics. The differential algebraic equations for the payload's linear acceleration are found by rearranging the definition of force $\mathbf{F} = m\mathbf{a}$ to isolate the second derivatives of the payload's states with respect to time $(\ddot{x}_{com}, \ddot{y}_{com}, \ddot{z}_{com})$:

$$\begin{bmatrix} \ddot{x}_{com} \\ \ddot{y}_{com} \\ \ddot{z}_{com} \end{bmatrix} = \begin{bmatrix} \frac{1}{m_{pl}} \sum_{i=1}^{3} (F_i \tan(\alpha_i)) \\ \frac{1}{m_{pl}} (\sum_{i=1}^{3} (F_i \tan(\beta_i)) \\ \frac{1}{m_{pl}} (\sum_{i=1}^{3} F_i) - g \end{bmatrix} \tag{3.10}$$

The linear accelerations of each attachment point are equal at all times due to the assumption of payload rigidity:

$$\ddot{x}_{com} = \ddot{x}_{c1} = \ddot{x}_{c2} = \ddot{x}_{c3} \tag{3.11}$$
$$\ddot{y}_{com} = \ddot{y}_{c1} = \ddot{y}_{c2} = \ddot{y}_{c3} \tag{3.12}$$
$$\ddot{z}_{com} = \ddot{z}_{c1} = \ddot{z}_{c2} = \ddot{z}_{c3} \tag{3.13}$$

Because of the assumption of cable inelasticity, the following cable length equalities must be enforced:

$$\sqrt{(x_{c1} - x_1)^2 + (y_{c1} - y_1)^2 + (z_{c1} - z_1)^2} = L \tag{3.14}$$
$$\sqrt{(x_{c2} - x_2)^2 + (y_{c2} - y_2)^2 + (z_{c2} - z_2)^2} = L \tag{3.15}$$
$$\sqrt{(x_{c3} - x_3)^2 + (y_{c3} - y_3)^2 + (z_{c3} - z_3)^2} = L \tag{3.16}$$

Owing to the collinear distribution of the attachment points about the center of mass, the following equalities involving the attachment points' separations hold:

$$r_{1,2} = \sqrt{(x_{c1} - x_{c2})^2 + (y_{c1} - y_{c2})^2 + (z_{c1} - z_{c2})^2} \tag{3.17}$$
$$r_{2,3} = \sqrt{(x_{c2} - x_{c3})^2 + (y_{c2} - y_{c3})^2 + (z_{c2} - z_{c3})^2} \tag{3.18}$$
$$r_{1,3} = \sqrt{(x_{c1} - x_{c3})^2 + (y_{c1} - y_{c3})^2 + (z_{c1} - z_{c3})^2} \tag{3.19}$$
$$r_{1,3} = r_{1,2} + r_{2,3} \tag{3.20}$$

We may therefore define an augmented state vector $\overline{X}$ and augmented input vector U for the system of the three agents and the payload:

$$\overline{X}_1 = [X_1^T, \dot{x}_{c1}^T, y_{c1}^T, \dot{y}_{c1}^T, z_{c1}^T, \dot{z}_{c1}^T]^T \tag{3.21}$$
$$\overline{X}_2 = [X_2^T, \dot{x}_{c2}^T, y_{c2}^T, \dot{y}_{c2}^T, z_{c2}^T, \dot{z}_{c2}^T]^T \tag{3.22}$$
$$\overline{X}_3 = [X_3^T, \dot{x}_{c3}^T, y_{c3}^T, \dot{y}_{c3}^T, z_{c3}^T, \dot{z}_{c3}^T]^T \tag{3.23}$$
$$\overline{X} = \begin{bmatrix} \overline{X}_1^T & \overline{X}_2^T & \overline{X}_3^T \end{bmatrix}^T \tag{3.24}$$
$$U_1 = \left[u_{x1}^T, u_{y1}^T, u_{z1}^T\right]^T \tag{3.25}$$
$$U_2 = \left[u_{x2}^T, u_{y2}^T, u_{z2}^T\right]^T \tag{3.26}$$
$$U_3 = \left[u_{x3}^T, u_{y3}^T, u_{z3}^T\right]^T \tag{3.27}$$
$$U = \begin{bmatrix} U_1^T & U_2^T & U_3^T \end{bmatrix}^T \tag{3.28}$$

We may evaluate the state transition equation for the system:

$$F(\overline{X}(t), U(t)) = \frac{d}{dt} \begin{bmatrix} x_1 \\ \dot{x}_1 \\ y_1 \\ \dot{y}_1 \\ z_1 \\ \dot{z}_1 \\ x_{c1} \\ \dot{x}_{c1} \\ y_{c1} \\ \dot{y}_{c1} \\ z_{c1} \\ \dot{z}_{c1} \\ x_2 \\ \dot{x}_2 \\ y_2 \\ \dot{y}_2 \\ z_2 \\ \dot{z}_2 \\ x_{c2} \\ \dot{x}_{c2} \\ y_{c2} \\ \dot{y}_{c2} \\ z_{c2} \\ \dot{z}_{c2} \\ x_3 \\ \dot{x}_3 \\ y_3 \\ \dot{y}_3 \\ z_3 \\ \dot{z}_3 \\ x_{c3} \\ \dot{x}_{c3} \\ y_{c3} \\ \dot{y}_{c3} \\ z_{c3} \\ \dot{z}_{c3} \end{bmatrix} = \begin{bmatrix} \dot{x}_1 \\ \frac{1}{m_{ag}}(u_{x1} - F_1 \tan(\alpha_1)) \\ \dot{y}_1 \\ \frac{1}{m_{ag}}(u_{y1} - F_1 \tan(\beta_1)) \\ \dot{z}_1 \\ \frac{1}{m_{ag}}(u_{z1} - F_1 - m_{ag}g) \\ \dot{x}_{c1} \\ \frac{1}{m_{pl}} \sum_{i=1}^{3}(F_i \tan(\alpha_i)) \\ \dot{y}_{c1} \\ \frac{1}{m_{pl}} \sum_{i=1}^{3}(F_i \tan(\beta_i)) \\ \dot{z}_{c1} \\ \frac{1}{m_{pl}}(\sum_{i=1}^{3} F_i) - g \\ \dot{x}_2 \\ \frac{1}{m_{ag}}(u_{x2} - F_2 \tan(\alpha_2)) \\ \dot{y}_2 \\ \frac{1}{m_{ag}}(u_{y2} - F_2 \tan(\beta_2)) \\ \dot{z}_2 \\ \frac{1}{m_{ag}}(u_{z2} - F_2 - m_{ag}g) \\ \dot{x}_{c2} \\ \frac{1}{m_{pl}} \sum_{i=1}^{3}(F_i \tan(\alpha_i)) \\ \dot{y}_{c2} \\ \frac{1}{m_{pl}} \sum_{i=1}^{3}(F_i \tan(\beta_i)) \\ \dot{z}_{c2} \\ \frac{1}{m_{pl}}(\sum_{i=1}^{3} F_i) - g \\ \dot{x}_3 \\ \frac{1}{m_{ag}}(u_{x3} - F_3 \tan(\alpha_3)) \\ \dot{y}_3 \\ \frac{1}{m_{ag}}(u_{y3} - F_3 \tan(\beta_3)) \\ \dot{z}_3 \\ \frac{1}{m_{ag}}(u_{z3} - F_3 - m_{ag}g) \\ \dot{x}_{c3} \\ \frac{1}{m_{pl}} \sum_{i=1}^{3}(F_i \tan(\alpha_i)) \\ \dot{y}_{c3} \\ \frac{1}{m_{pl}} \sum_{i=1}^{3}(F_i \tan(\beta_i)) \\ \dot{z}_{c3} \\ \frac{1}{m_{pl}}(\sum_{i=1}^{3} F_i) - g \end{bmatrix} \tag{3.29}$$

3.2 Problem Statement

We may now formulate the problem statement for centralized CPT using a cable-suspended system of agents. Consider N agents governed by the system dynamics in (3.29) and satisfying the cable angle dynamics in (3.1)–(3.6), the cable tension equalities in (3.7)–(3.9), the cable length equalities in (3.14)–(3.16), and the attachment point separation equalities in (3.17)–(3.20). Let each mission have a fixed duration t_f. We assume that over the time interval $[0, t_f]$ the system has the initial state $\overline{X}(0)$ and target state $\overline{X}(t_f)$, and that the agents communicate their states to a central controller with negligible delay. Define a function $\omega : \mathbb{R}^3 \to \mathbb{R}$ corresponding to the height clearance between the payload's center of mass and a fixed obstacle. The problem requires the design of a feedback control law such that:

- the agents' and payload's poses converge to the target state over the time interval $[0, t_f]$: $\lim_{t \to t_f} \overline{X}_i(t) = \overline{X}_i(t_f)\ \forall i \in \{1, ..., N\}$;
- the sum of each agent's squared thrusts should be minimized;
- each agent remains connected to the payload, i.e. given the cable length $L > 0$ $\sqrt{(x_{ci}(t) - x_i(t))^2 + (y_{ci}(t) - y_i(t))^2 + (z_{ci}(t) - z_i(t))^2} = L,\ \forall i \in \{1, ..., N\},\ \forall t \geq 0$;
- inter-agent collisions are avoided, i.e. given a minimum separation $\sigma > 0$, $\sqrt{(x_i(t) - x_j(t))^2 + (y_i(t) - y_j(t))^2 + (z_i(t) - z_j(t))^2} > \sigma,\ \forall i, j \in \{1, ..., N\},\ i \neq j,\ \forall t \geq 0$;
- collisions between the payload and the obstacle are avoided, i.e. given a minimum clearance $\kappa > 0$, $\omega(x_{ci}(t), y_{ci}(t), z_{ci}(t)) > \kappa,\ \forall i \in \{1, ..., N\},\ \forall t \geq 0$.

3.3 Controller Architecture

To implement the feedback control laws described in the problem statement, we must specify the optimal control problem for the system. Assume that the system has a discrete-time controller with sampling period $t_s > 0$. Consider a sequence of sample times $\{t_k\}_{k \in \mathbb{N}}$ such that $t_k \in [0, t_f]$ and $t_{k+1} = t_k + t_s$. At sample time t_k the controller uses its knowledge of the system's state to solve the following optimal control problem:

$$\min_{U(t)} \left\| \overline{X}(t_k) - \overline{X}(t_f) \right\|^2 + \sum_{\tau = t_k}^{t_f} \left\| U(\tau) \right\|^2 \tag{3.30a}$$

subject to

$$\sqrt{(x_{ci}(t) - x_i(t))^2 + (y_{ci}(t) - y_i(t))^2 + (z_{ci}(t) - z_i(t))^2} = L,\ \forall i \in \{1, ..., N\},\ \forall t \in [0, t_f], \tag{3.30b}$$

$$\sqrt{(x_i(t) - x_j(t))^2 + (y_i(t) - y_j(t))^2 + (z_i(t) - z_j(t))^2} > \sigma,\ \forall i, j \in \{1, ..., N\},\ i \neq j,\ \forall t \in [0, t_f], \tag{3.30c}$$

$$\omega(x_{ci}(t), y_{ci}(t), z_{ci}(t)) > \kappa,\ \forall i \in \{1, ..., N\},\ \forall t \in [0, t_f]. \tag{3.30d}$$

We have chosen a decreasing-horizon optimal control problem (DHOCP) formulation; the lower bound t_k of the summation in (3.30) converges to the final time t_f during the mission. We require a DHOCP (as opposed to a fixed-horizon optimal control problem) so that the system reaches the specified terminal state by the end of the mission. When using a fixed number of collocation points for the optimization, the decreasing optimization horizon $(t_f - t_k)$ leads to smaller gaps between the predicted samples of $U(\tau)$, known as a finer optimization mesh.

We may now transcribe the DHOCP given by (3.30) into a format compatible with the ICLOCS2 trajectory optimization package.

Chapter 4

Implementation

In simulating the controller's performance, we have relied on the ICLOCS2 trajectory optimization package for solving the DHOCP within the MATLAB-Simulink environment. Our approach has involved creating basic mathematical models in code and adding constraints incrementally to converge to the system described in Chapter 3. We have constructed four successive models for this project by:

- relocating a single agent;
- relocating a single agent attached via a cable to a payload;
- relocating two agents, each attached via a cable to the payload;
- relocating three agents, each attached via a cable to the payload.

We subsequently describe the structural features of each model and how each model's performance has been simulated.

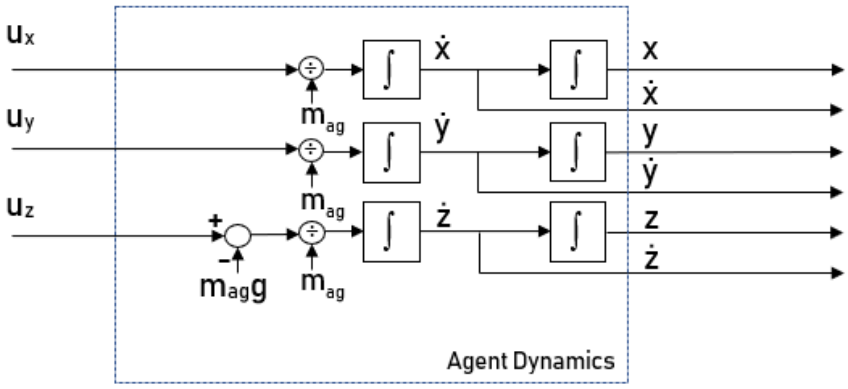


Figure 4.1: Double-Integrator System for a Point Mass in $\mathbb{R}^3$

4.1 Single-Agent Relocation

We begin by considering the problem of relocating a single agent from one position to another. We approximate the agent by a point mass and consider the translational dynamics of a double-integrator system in three dimensions (Figure 4.1). Let this system have six states $(x, \dot{x}, y, \dot{y}, z, \dot{z})$ and three inputs (u_x, u_y, u_z).

Representing the system state with $\hat{X}(t)$, we can formulate a DHOCP similar to (3.30):

$$\min_{u_x(\tau),u_y(\tau),u_z(\tau)} \left\| \hat{X}(t_k) - \hat{X}(t_f) \right\|^2 + \sum_{\tau=t_k}^{t_f} \|u_x(\tau) + u_y(\tau) + u_z(\tau)\|^2 \tag{4.1a}$$

subject to

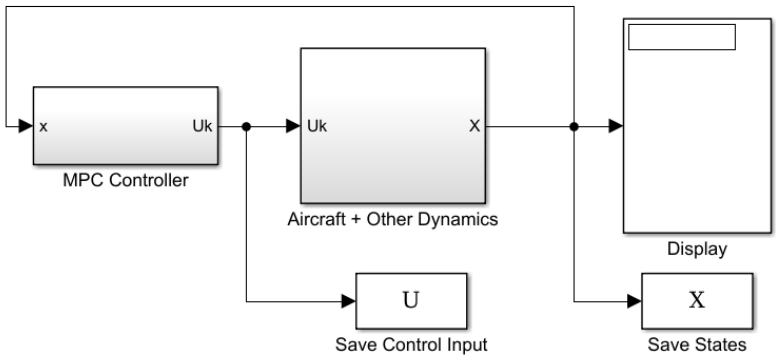
$$z(t) - h(\tanh(x(t) - x_a) - \tanh(x(t) - x_b)) > \kappa, \ \forall t \in \mathbb{R}^+ \tag{4.1b}$$

where the obstacle's height $h = 1.00\,\text{m}$, the clearance $\kappa = 0.50\,\text{m}$, and the parameters $x_a = 3.10\,\text{m}$ and $x_b = 4.10\,\text{m}$ correspond respectively to the start and end of the obstacle's length in the x-axis.

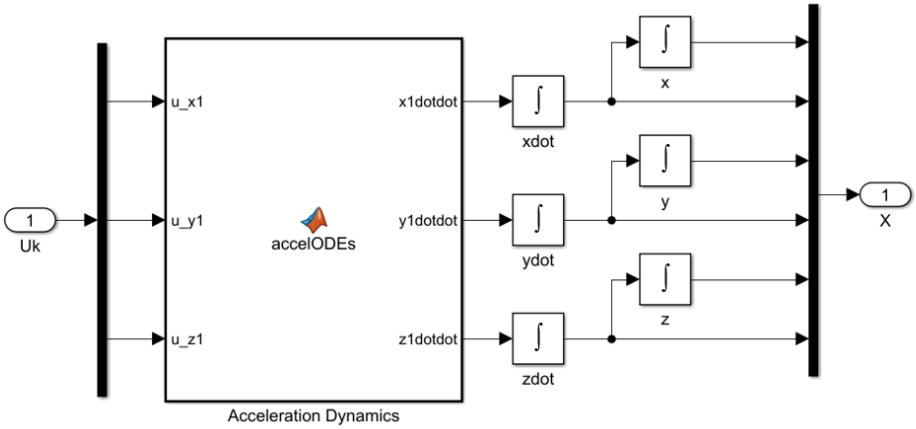
We may then transcribe (4.1) into a format appropriate for ICLOCS2 (a 'plant' file containing the state transition equations, and a 'problem' file containing all information about the DHOCP). Note that to reduce the computational burden on the solver we have fixed the end time t_f and the number of mesh points for the solver, and set the obstacle function $\omega(\cdot)$ equal to a composite function that is smooth, continuous and infinitely differentiable over its entire domain.

There are two methods in which we may use the output from ICLOCS2. In open-loop simulations the DHOCP is fed to the optimization solver once, which then searches for the optimal sequence of control inputs. This requires a script in MATLAB to initiate the solver with the problem settings, perform automatic mesh refinement, and plot the evolution of the system fed by the generated control inputs.

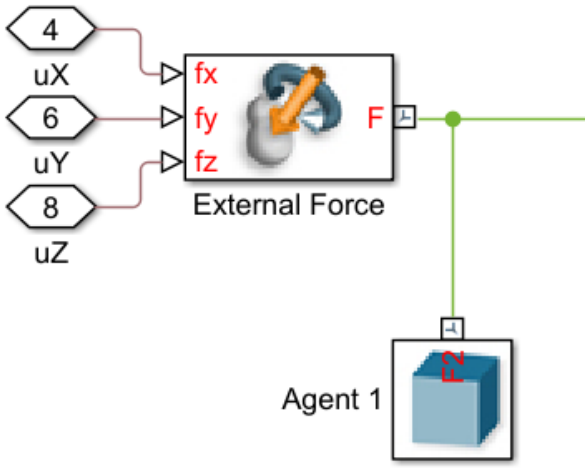
In closed-loop simulations the solver is embedded into a control block in Simulink and is fed live positional data from a system model driven by the solver's output (Figure 4.2a). The system model may be in the form of an ADM model (Figure 4.2b) or a GDM model using the Simscape Multibody library in Simulink (Figure 4.2c).



(a) Top-Level Simulink Diagram



(b) ADM Model using Ordinary Differential Equations



(c) GDM Model of Single Agent in Simscape

Figure 4.2: Closed-Loop Simulation Schematic Diagrams

4.2 Single-Agent Payload Relocation

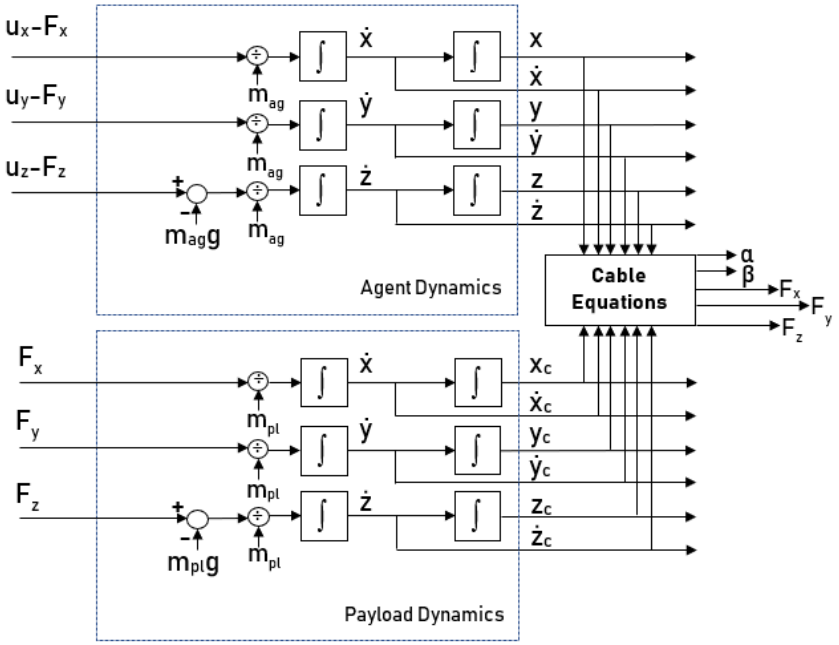


Figure 4.3: Double-Integrator System for Agent and Cable-Suspended Payload in $\mathbb{R}^3$

Next we consider the problem of relocating a single agent with a cable-suspended payload. We approximate the agent and payload by point masses joined by a taut cable, so we may consider the translational dynamics of two double-integrator systems in three dimensions (Figure 4.3). Let this system have twelve states $(x, \dot{x}, y, \dot{y}, z, \dot{z}, x_c, \dot{x}_c, y_c, \dot{y}_c, z_c, \dot{z}_c)$ and three inputs (u_x, u_y, u_z). We may define $\{F_x, F_y, F_z\}$ as the components of the cable tension resolved in the x-, y-, and z-axes, and α_1 and β_1 as per (3.1) and (3.2) respectively.

Representing the system state with $\widetilde{X}(t)$, we can formulate a DHOCP and transcribe this into code for ICLOCS2:

$$\min_{u_x(\tau),u_y(\tau),u_z(\tau)} \left\| \widetilde{X}(t_k) - \widetilde{X}(t_f) \right\|^2 + \sum_{\tau=t_k}^{t_f} \|u_x(\tau) + u_y(\tau) + u_z(\tau)\|^2 \tag{4.2a}$$

subject to

$$z(t) - h(\tanh(x(t) - x_a) - \tanh(x(t) - x_b)) > \kappa, \ \forall t \in \mathbb{R}^+ \tag{4.2b}$$

$$\sqrt{(x_c(t) - x(t))^2 + (y_c(t) - y(t))^2 + (z_c(t) - z(t))^2} = L, \ \forall t \in \mathbb{R}^+ \tag{4.2c}$$

where the cable length L=1.20 m, the obstacle's height $h = 1.00\,\text{m}$, the clearance $\kappa = 0.50\,\text{m}$, and the parameters $x_a = 3.10\,\text{m}$ and $x_b = 4.10\,\text{m}$ correspond respectively to the start and end of the obstacle's length in the x-axis. Note that in addition to the obstacle function we must impose an equality constraint to ensure that the agent and payload remain connected at all times.

As previously discussed we may run ICLOCS2 simulations in open-loop or closed-loop; the code is not substantially different, however the GDM model represents the payload as a point mass and the taut cable as a rigid cylinder with two spherical joints (Figure 4.4).

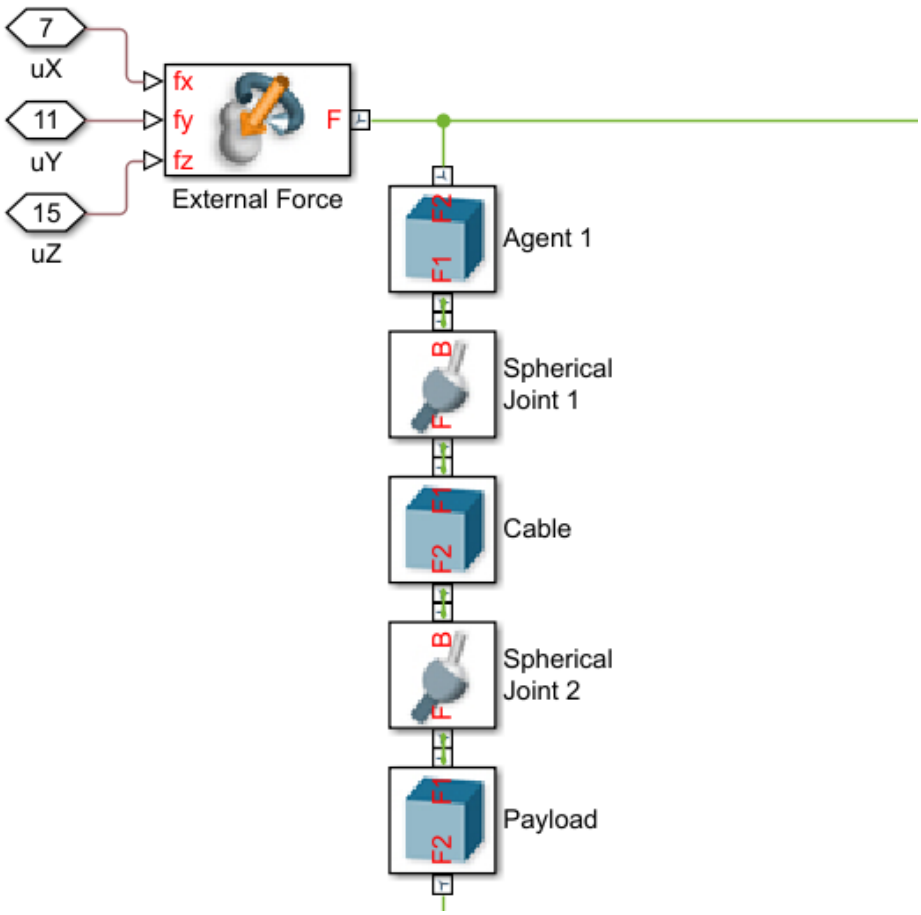


Figure 4.4: GDM Model of Single Agent with Cable-Suspended Payload in Simscape

4.3 Centralized Dual-Agent CPT

Next we may derive a model for the most basic CPT scheme using two agents. The DHOCP is identical to that of the single-agent scheme in 4.2, with the number of agents $N = 2$. Hence the ADM model of the system is similar to that of the single-agent scheme, augmented with extra states and inputs associated with the second agent and its payload:

$$f(\overline{X}(t), U(t)) = \frac{d}{dt}\begin{bmatrix} \overline{X}_1(t) \\ \overline{X}_2(t) \end{bmatrix} = \begin{bmatrix} f_1(t) \\ f_2(t) \end{bmatrix}, \tag{4.3a}$$

where

$$f_j = \begin{bmatrix} \dot{x}_j \\ \frac{1}{m_{ag}}(u_{xj} - F_j \tan(\alpha_j)) \\ \dot{y}_j \\ \frac{1}{m_{ag}}(u_{yj} - F_j \tan(\beta_j)) \\ \dot{z}_j \\ \frac{1}{m_{ag}}(u_{zj} - F_j - m_{ag}g) \\ \dot{x}_{cj} \\ \frac{1}{m_{pl}}\sum_{i=1}^{2}(F_i \tan(\alpha_i)) \\ \dot{y}_{cj} \\ \frac{1}{m_{pl}}\sum_{i=1}^{2}(F_i \tan(\beta_i)) \\ \dot{z}_{cj} \\ \frac{1}{m_{pl}}(\sum_{i=1}^{2} F_i) - g \end{bmatrix}, j \in \{1, 2\}. \tag{4.3b}$$

The GDM model is a constrained duplication of the single-agent system with a payload (Figure 4.5).

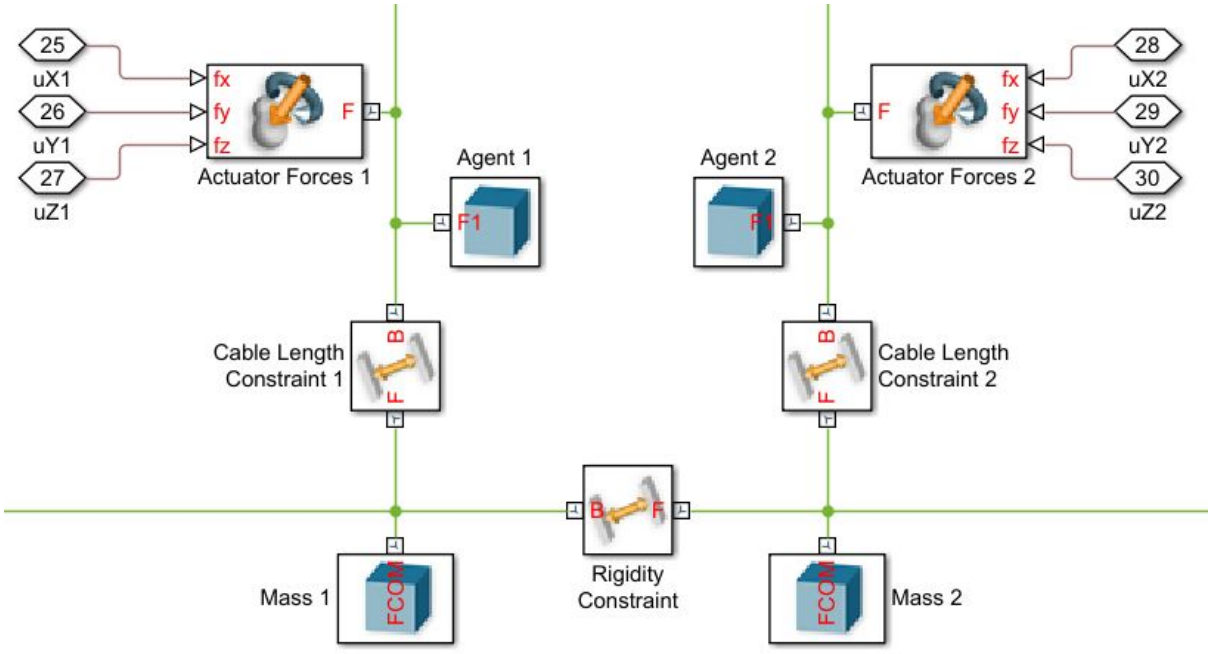


Figure 4.5: GDM Model of the Dual-Agent CPT System in Simscape

4.4 Centralized Triple-Agent CPT

We may construct the triple-agent system by a constrained triplication of the system of the single-agent system with a cable-suspended payload. Using the system dynamics specified in (3.29) and the DHOCP specified in (3.30), the code for ICLOCS2 can be transcribed in a similar manner to what has been previously described. Both the state vector and input vector are enlarged proportionally to the number of agents, while additional path constraints are included to maintain agent separation and the payload's rigidity. Due to the increased computational complexity of the DHOCP, we have adapted the GDM model in Section 4.2. The updated model splits the payload into three points of equal mass, each of which is connected to an agent and each other via distance constraints (Figure 4.6).

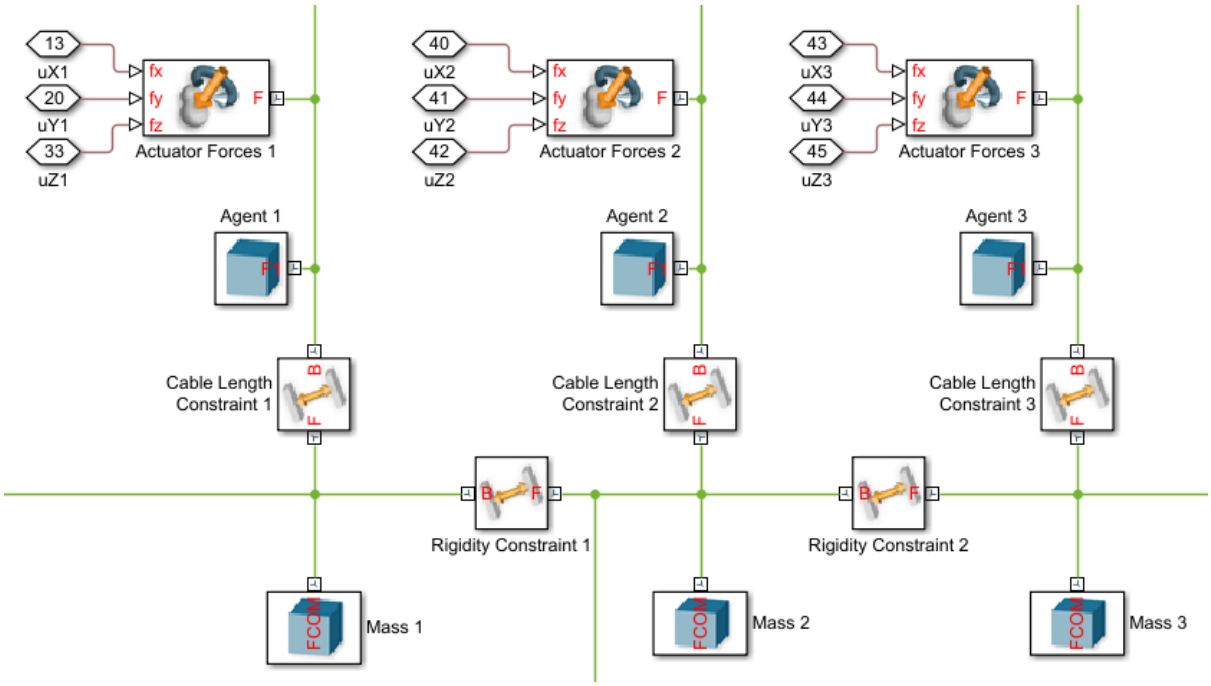


Figure 4.6: GDM Model of the Triple-Agent CPT System in Simscape

Chapter 5

Results

In this chapter we present and discuss the outcomes of the simulations described in Chapter 4. For each simulation we have assumed the agent's mass $m_{ag} = 810\,\mathrm{g}$ and the payload's mass $m_{pl} = 2.00\,\mathrm{kg}$.

5.1 Single-Agent Relocation

5.1.1 Open-Loop Simulations

Error Analysis: We begin with the outcomes of the open-loop simulations. To determine the most appropriate solver settings for ICLOCS2, we have conducted a series of tests with three common transcription methods (the Euler, Trapezoidal, and Hermite-Simpson methods in increasing order) and different numbers of mesh points {5,10,15,20,25,40,50,60,80}. With each combination of transcription method and number of mesh points we have performed the open-loop simulation and extracted the computation time (Figure 5.1) and maximum error (Figure 5.2).

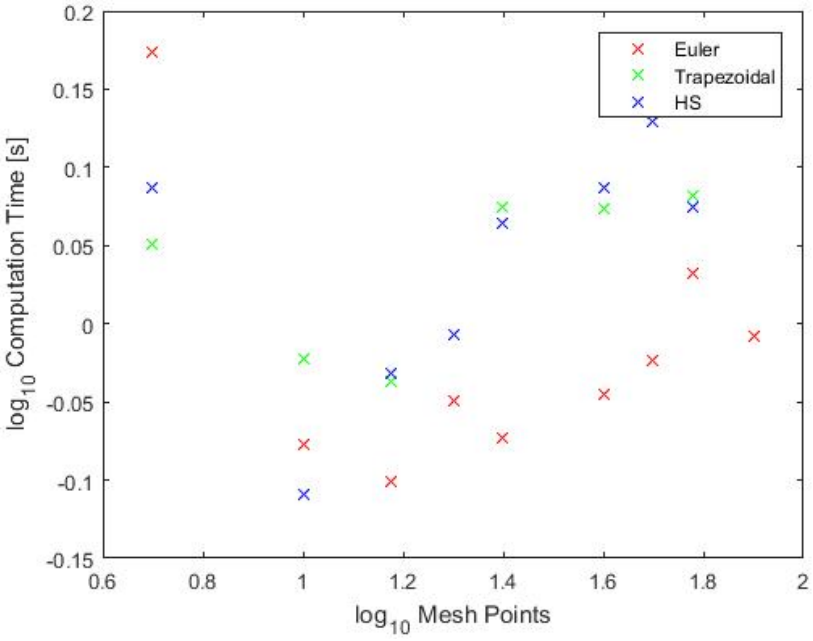


Figure 5.1: Computation Time (Single Agent, Open Loop)

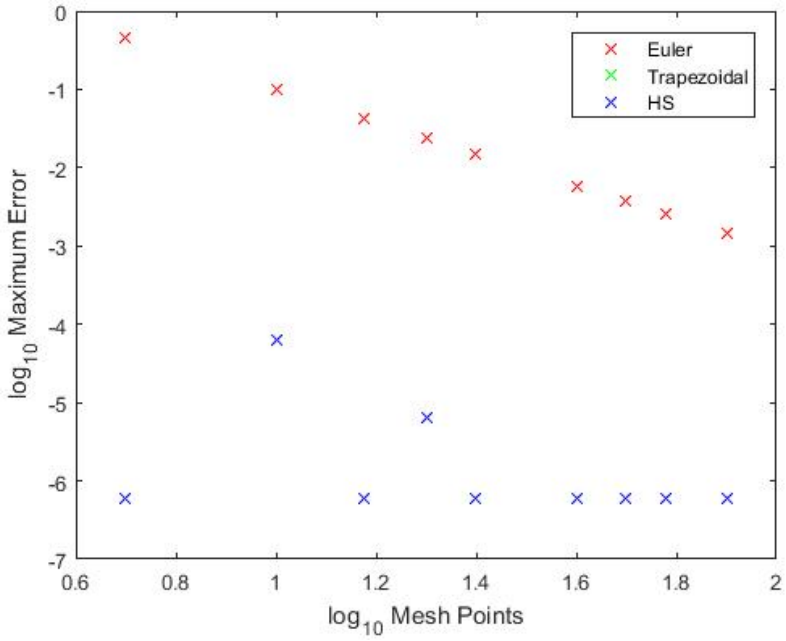


Figure 5.2: Maximum Error (Single Agent, Open Loop)

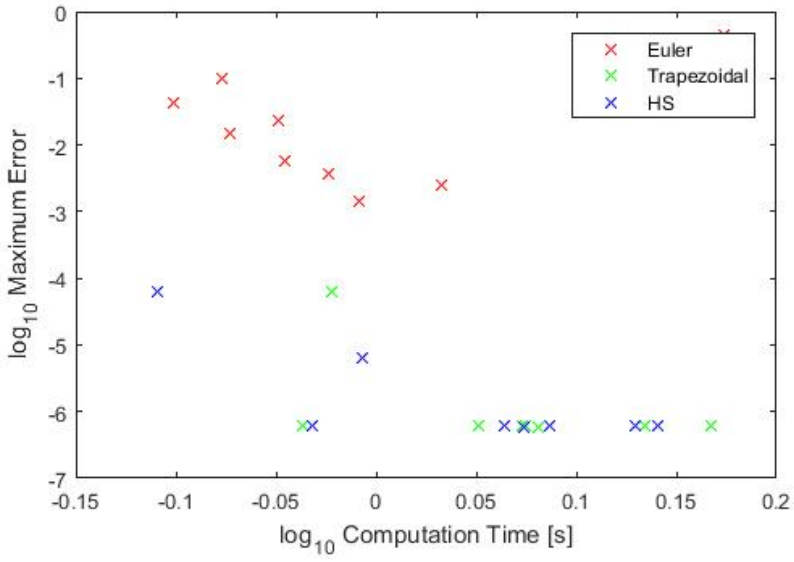


Figure 5.3: Maximum Error v. Computation Time (Single Agent, Open Loop)

Plotting the computation time against maximum error yields the graph in Figure 5.3. In these three graphs we notice a number of phenomena:

- Computation time grows by increasing the order of transcription method or the number of mesh points (Figure 5.1);
- The maximum error decreases by increasing the order of the transcription method or the number of mesh points (Figure 5.2);
- For each transcription method a Pareto optimality boundary appears in Figure 5.3, such that we cannot reduce one variable without creating an increase in the other variable.

Solver Parameter Selection: We have used the graph in Figure 5.3 to select the combination of transcription method and number of mesh points that yields the fastest computation time with a suitably low error. A Hermite-Simpson transcription method with 10 mesh points runs in 794 ms. We have employed these parameters to solve the DHOCP described in Section 4.1, yielding the evolutions of the system states (Figure 5.4) and the input sequence (Figure 5.5). At all times the computed solution complies with all constraints on the state and input values.

Physical Interpretation: We first notice that the system navigates in the presence of a spatial obstacle in the X-Z plane by increasing thrust in the z-axis above the hovering thrust (equal to $m_{ag}g = 0.81 * 9.81 = 7.95\,\mathrm{N}$) in order to accelerate upwards. The thrust steadily decreases until the agent reaches the peak of the obstacle, at which the z velocity changes from positive to negative and begins freefall. The thrust then increases to counteract the agent's freefall, bringing the agent to the target height at the end of the mission.

Thrust in the x-axis is initially positive, causing the agent to accelerate. The thrust then decreases during the mission to slow the agent down, crossing from positive to negative as soon as the agent reaches the midway point. The x velocity consequently decreases until the agent stops at the target position.

Thrust in the y-axis is negligible, as desired. The non-zero input values can be attributed to numerical errors from the optimization solver.

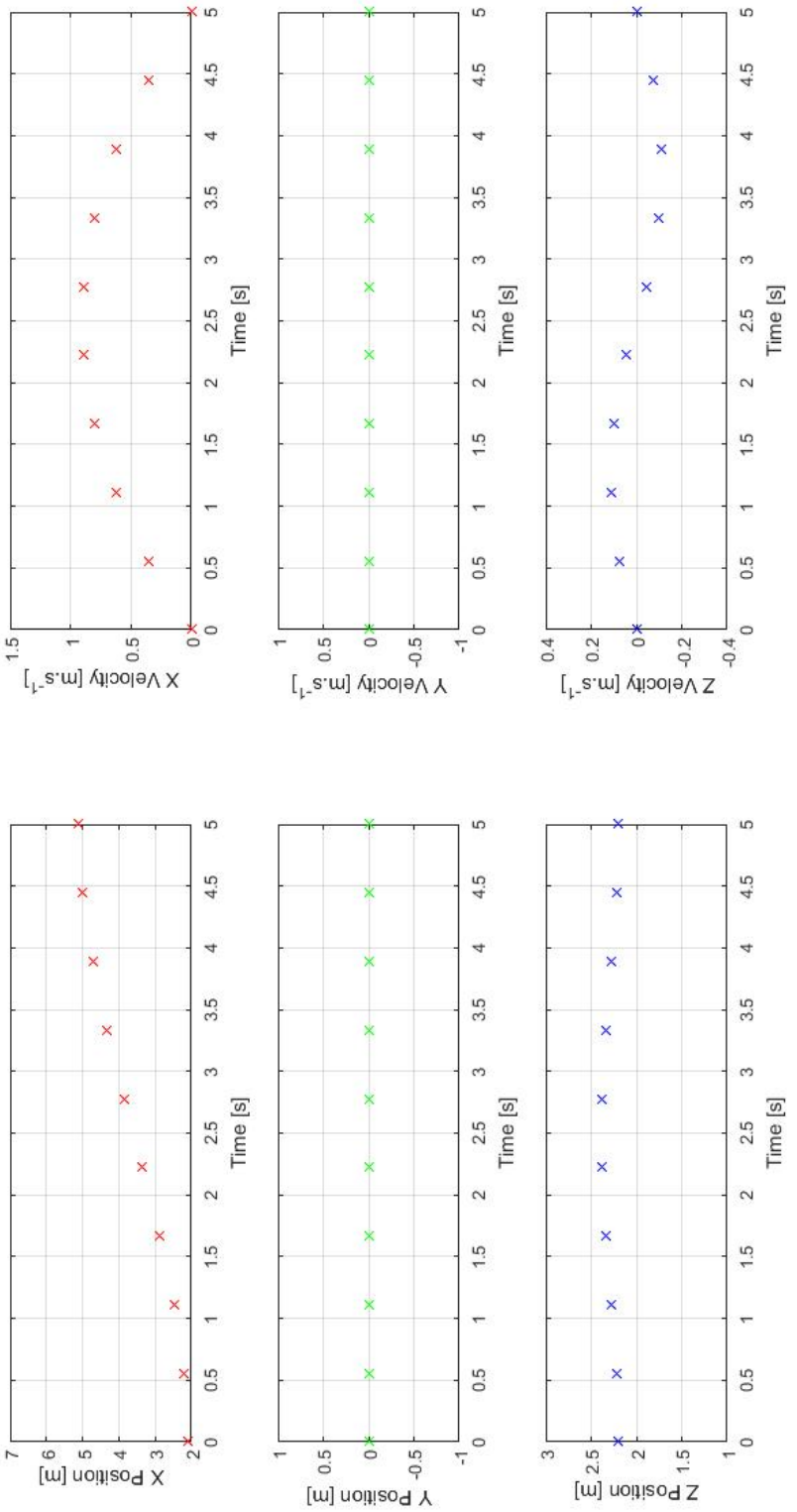


Figure 5.4: Evolution of System States (Single Agent, Open Loop)

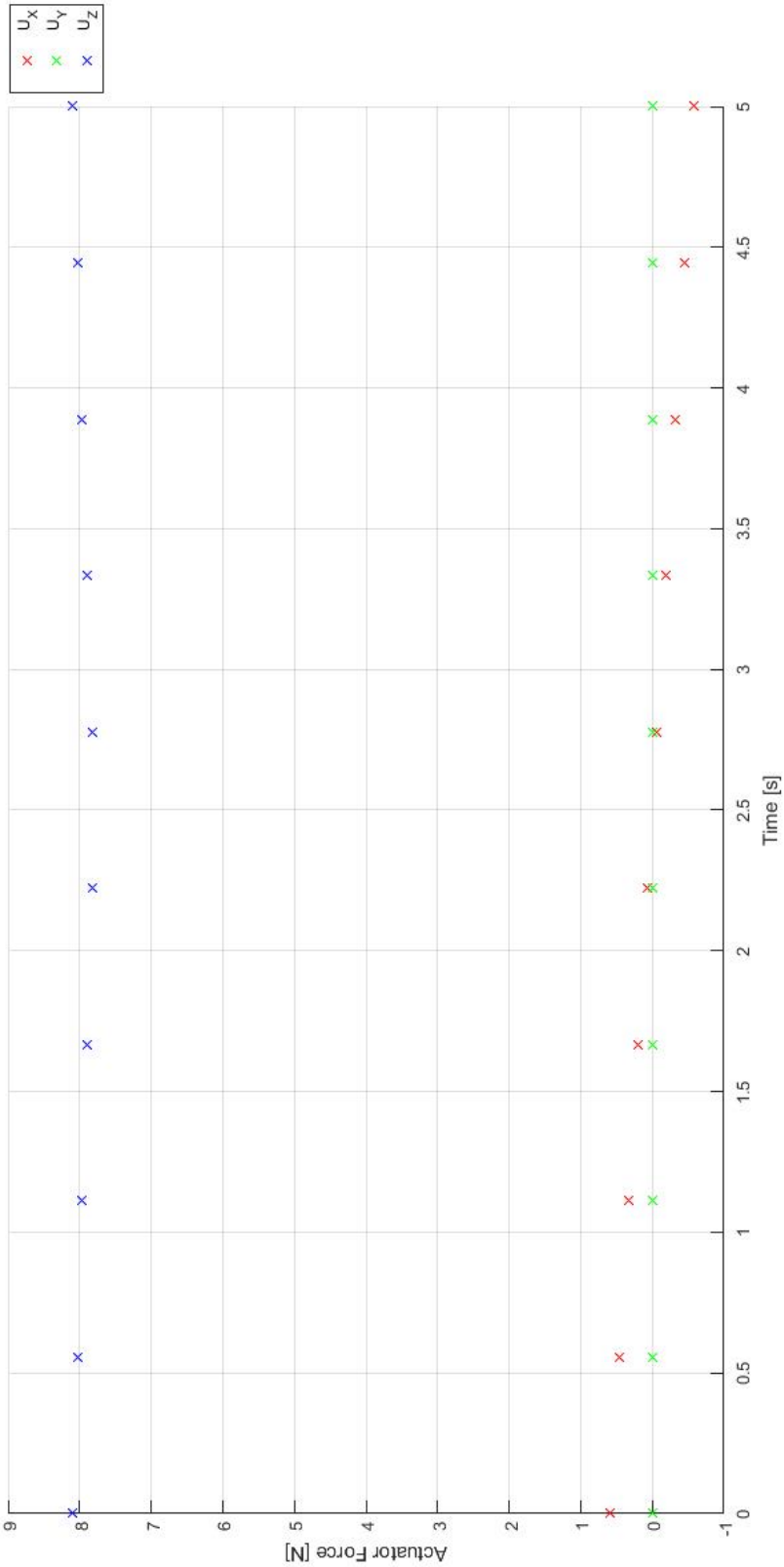


Figure 5.5: Evolution of System Inputs (Single Agent, Open Loop)

5.1.2 Closed-Loop Simulations with the ADM and GDM Models

Implementation: We have implemented the MPC-based algorithm described in Section 4.1, using either the ADM model or the GDM model of the single agent in closed-loop with the controller. Using the same solver parameters as for the open-loop simulation and reoptimizing the inputs every second yields the closed-loop evolutions of the system states (Figures 5.6 and 5.7) and the input sequence (Figure 5.8 and 5.9).

Observations: We notice that these graphs are consistent with those of the open-loop simulation, with a greater number of mesh points and greater numerical errors in u_y. The computational runtime of the entire closed-loop simulation using the ADM model (20.2 s) is comparable to that of the GDM model (17.4 s) and is an order of magnitude slower than that of the open-loop simulation (794 ms). This is likely due to overhead delays introduced by Simulink in performing tasks such as:

- compiling the diagram,
- initializing model variables,
- numerical integration of model variables.

The latter task is a necessary feature of all closed-loop simulations since we wish to observe how the system behaves in response to the controller's generated inputs, as opposed to the evolution of the system states predicted by ICLOCS2 in open-loop simulations. For numerical integration in Simulink we have chosen to use the MATLAB function *ode23t*, a variable-step solver that can handle the differential algebraic equations contained in (3.29) [38].

In Figures 5.6 and 5.7 we may observe that the evolutions of the system's states corresponding to the closed-loop simulations using the ADM and GDM models are identical, with both figures resembling an interpolation of the open-loop system states in Figure 5.4. A similar situation holds for the closed-loop evolutions of the system inputs in Figures 5.8 and 5.9, which resemble an interpolation of the open-loop system inputs in Figure 5.5.

Mission Visualization: The closed-loop simulation using the GDM model can visualize the system's behavior, representing the data in Figure 5.7 graphically. A series of still images from the mission visualization are depicted in Figure 5.10. In order to make the agent's height changes more obvious we have superimposed a line on each image representing the initial height.

Consistent with the evolution of the agent's x and z positions in Figure 5.7, the agent progresses in the $+x$ direction between Figures 5.10a–5.10f. We may infer that the agent accelerates in the x-axis from the differences in the agent's x position between Figures 5.10a–5.10b, and Figures 5.10b–5.10c. We may also infer that the agent decelerates in the x-axis from the differences in the agent's x position between Figures 5.10d–5.10e, and Figures 5.10e–5.10f. These behaviors are consistent with the rise and fall of the agent's x velocity in Figures 5.6 and 5.7. Using similar arguments, we can deduce an increase and decrease in the agent's z position and z velocity, consistent with obstacle avoidance in Figures 5.6 and 5.7.

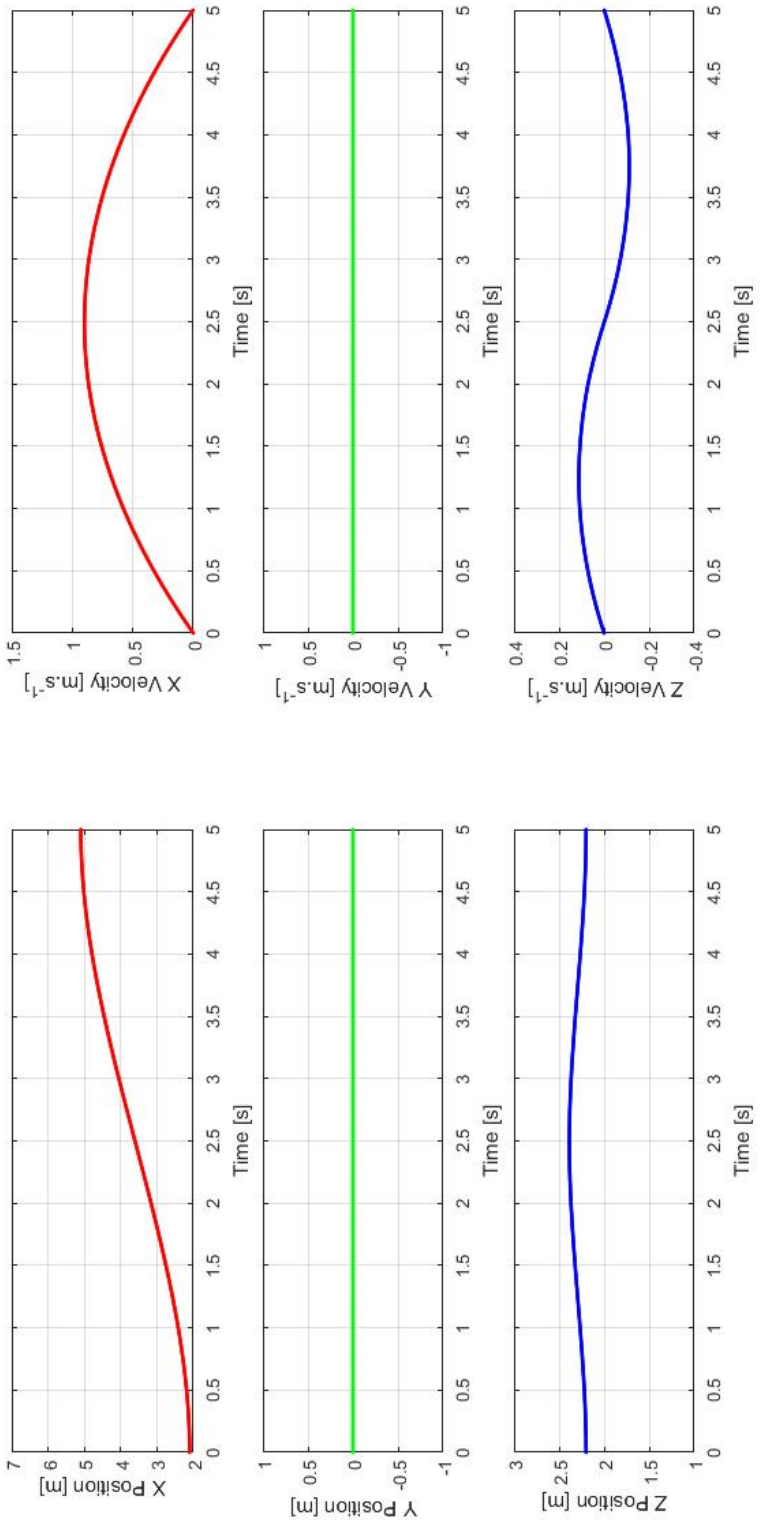


Figure 5.6: Evolution of System States (Single Agent, Closed Loop with ADM Model)

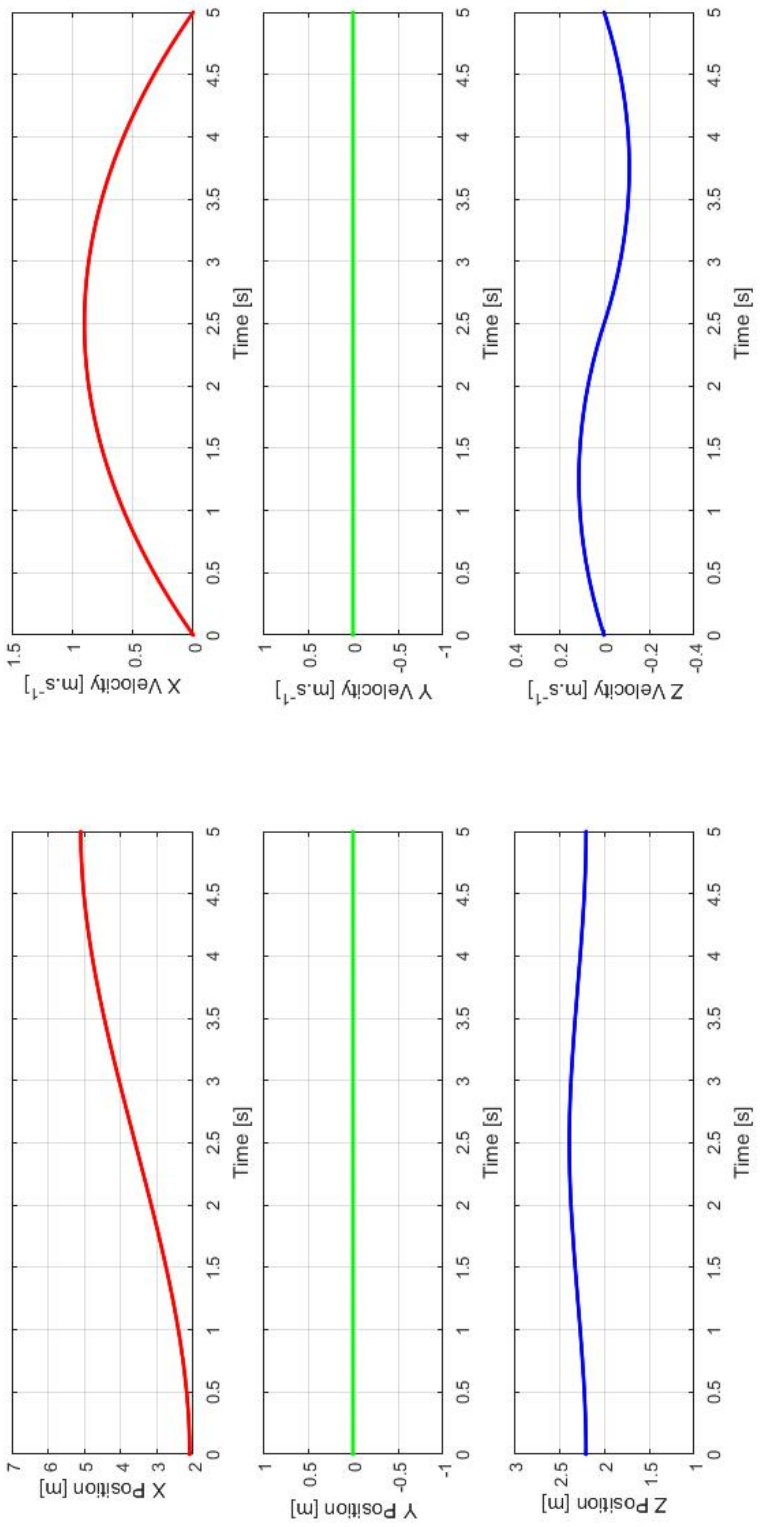


Figure 5.7: Evolution of System States (Single Agent, Closed Loop with GDM Model)

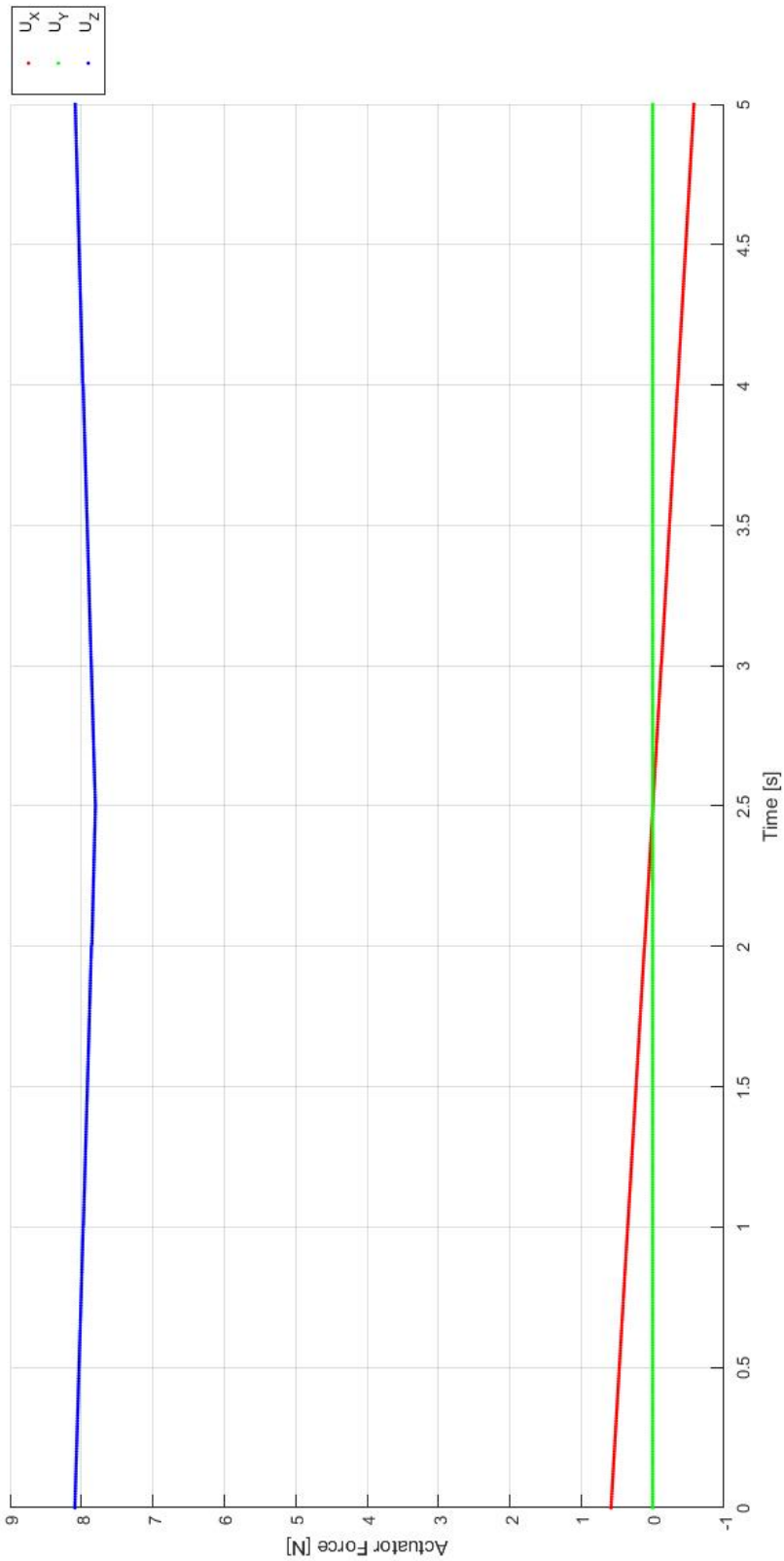


Figure 5.8: Evolution of System Inputs (Single Agent, Closed Loop with ADM Model)

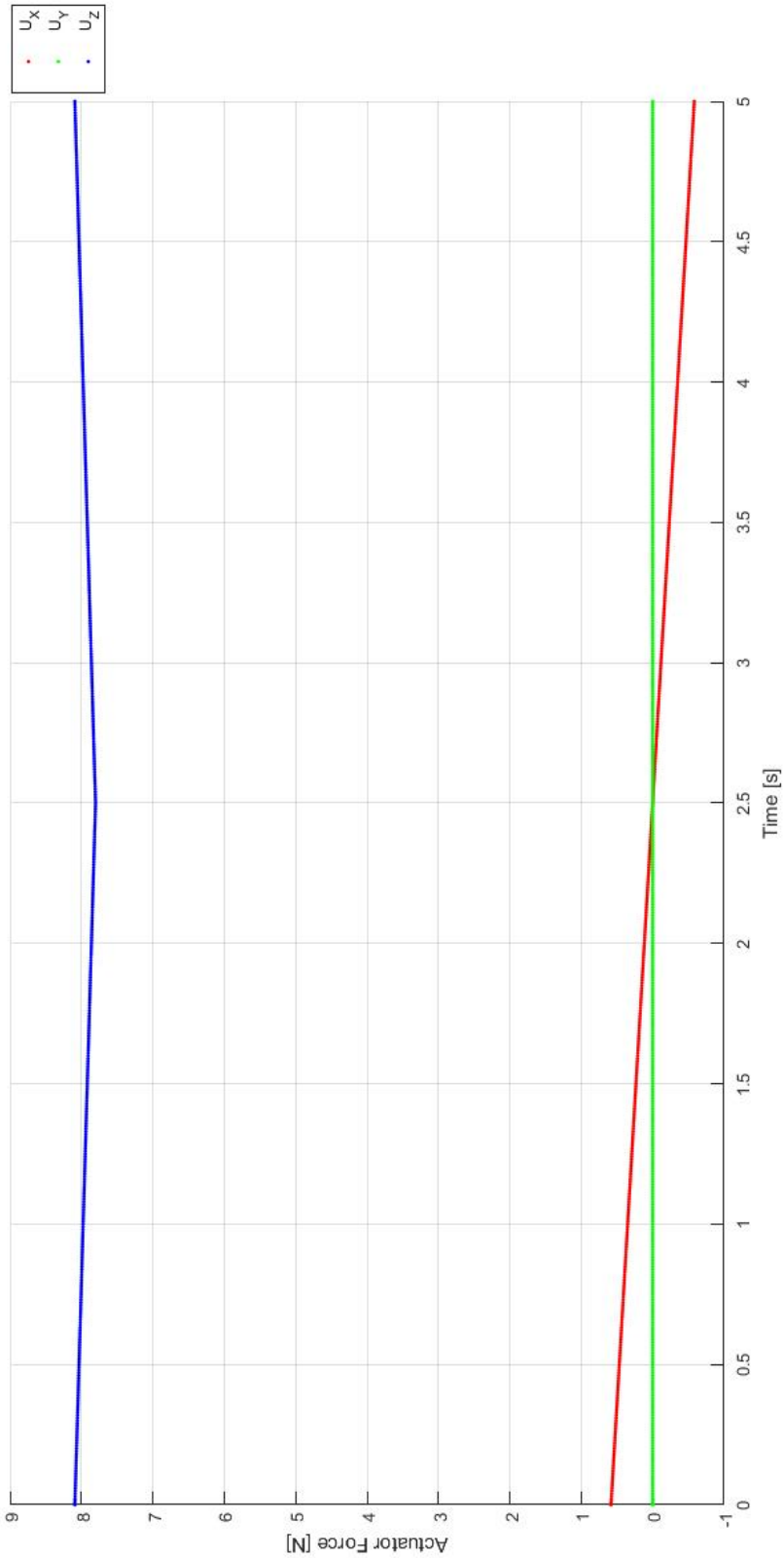
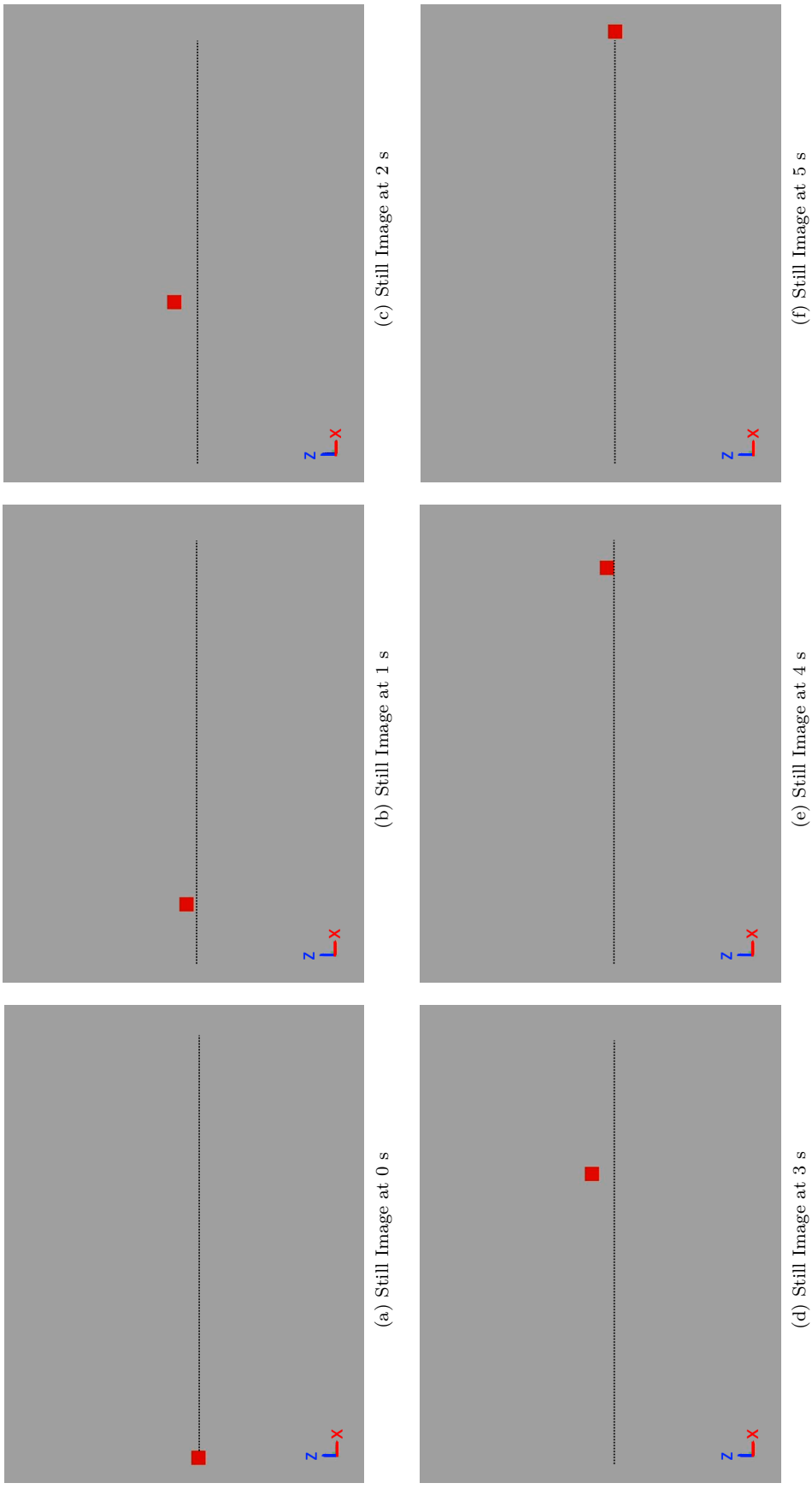


Figure 5.9: Evolution of System Inputs (Single Agent, Closed Loop with GDM Model)



(a) Still Image at 0 s (b) Still Image at 1 s (c) Still Image at 2 s

(d) Still Image at 3 s (e) Still Image at 4 s (f) Still Image at 5 s

Figure 5.10: Mission Visualization (Single Agent, Closed Loop with GDM Model)

5.2 Single-Agent Payload Relocation

We consider a single agent with a cable-suspended payload in this section.

5.2.1 Open-Loop Simulations

Error Analysis: As in the previous section, we begin with the outcomes of the open-loop simulations. We have conducted the same series of tests with combinations of the three transcription methods (the Euler, Trapezoidal, and Hermite-Simpson methods) with different numbers of mesh points {5,10,15,20,25,40,50,60,80}. We have performed the open-loop simulations for each combination and extracted the computation time (Figure 5.11) and maximum error (Figure 5.12).

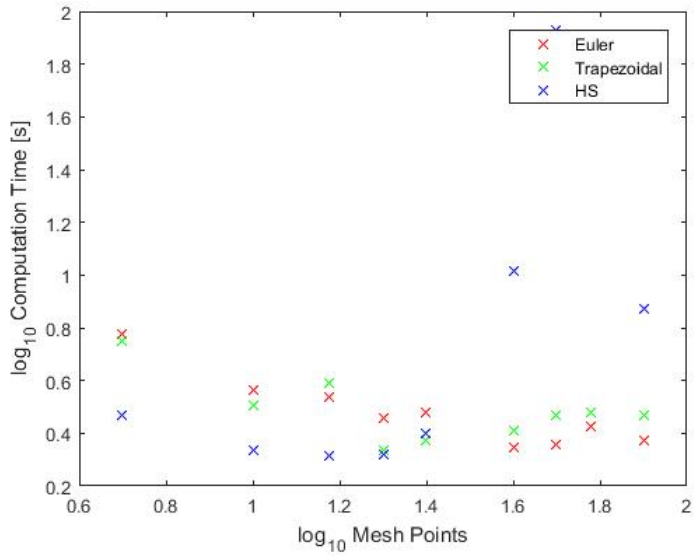


Figure 5.11: Computation Time (Single Agent with Payload, Open Loop)

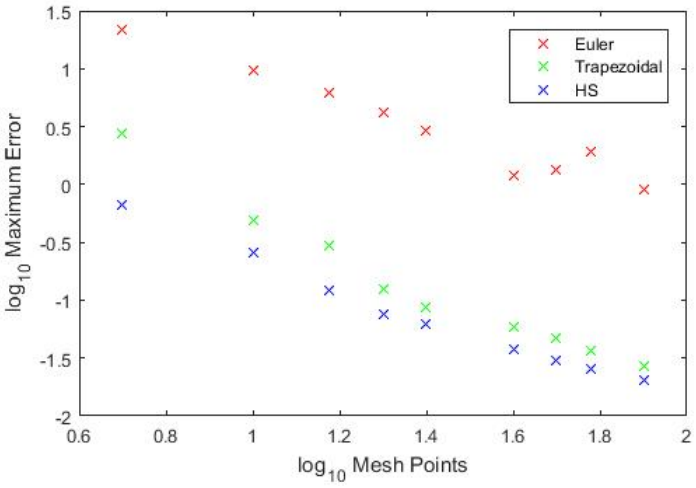


Figure 5.12: Maximum Error (Single Agent with Payload, Open Loop)

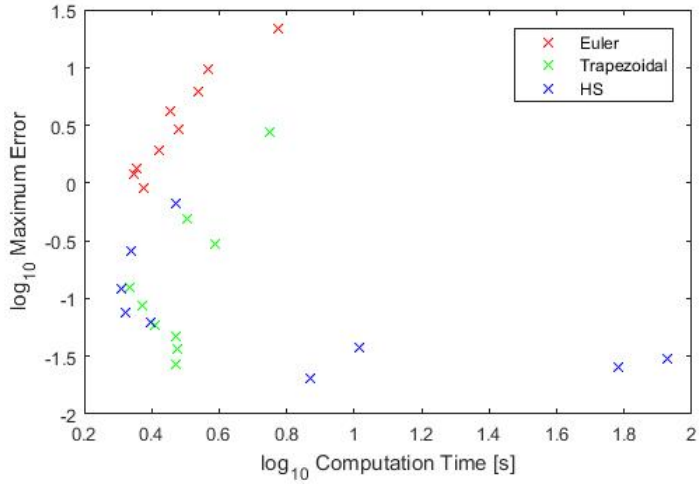


Figure 5.13: Maximum Error v. Computation Time (Single Agent with Payload, Open Loop)

Plotting the computation time against maximum error yields the graph in Figure 5.13. In these three graphs we notice the same phenomena as in Section 5.1.1, with longer computation times in Figure 5.11 than those of the single-agent system due to the increased complexity of the DHOCP.

Solver Parameter Selection: We have selected the combination of transcription method and number of mesh points that yields the fastest computation time with a suitably low error. A Hermite-Simpson transcription method with 15 mesh points runs in 1.99 s. We have then used these parameters to solve the DHOCP (including a spatial obstacle) described in Section 4.2, yielding the evolutions of the system states (Figure 5.14) and the input sequence (Figure 5.15). At all times the computed solution complies with all constraints on the state and input values.

Physical Interpretation: In Figure 5.14 we notice that the system navigates in the presence of a spatial obstacle in the X-Z plane by varying the thrust in the x- and z-axes. Initially the thrust in the z-axis is increased above the hovering thrust (equal to $(m_{ag}+m_{pl})g = 2.81*9.81 = 27.6\,\text{N}$) to cause a net upward acceleration. Likewise as a proxy of u_z, F initially exceeds the hovering thrust for the payload (equal to $m_{pl}g = 2*9.81 = 19.6\,\text{N}$) to cause a net upward acceleration. Both u_z and F steadily decrease until the agent reaches the peak of the obstacle, at which the z velocities of the agent and payload changes from positive to negative and both enter freefall. Both u_z and F then increase to counteract freefall, bringing the agent and the payload to their respective target heights at the end of the mission.

Thrust in the x-axis is initially positive, causing the agent to accelerate. The thrust remains constant for approximately 750 ms to allow the agent and the payload to accelerate. We notice that the payload initially lags behind the agent, hence $\tan\alpha$ is positive. The thrust begins to decrease after 750 ms, causing the agent to slow down and the payload to temporarily lead the agent. The thrust crosses from positive to negative as soon as the agent reaches the midway point, causing the x velocities of the agent and payload to decrease. In this part of the mission the payload tends to lead the agent, hence $\tan\alpha$ is negative. At the end of the mission the agent briefly accelerates to counteract any overshoot from the payload, allowing both the agent and the payload to stop at the target position. Evidently the oscillations in the cable angle reflect the system's pendulum nature.

Thrust in the y-axis is negligible, as desired; the non-zero input values can be attributed to numerical errors from the optimization solver.

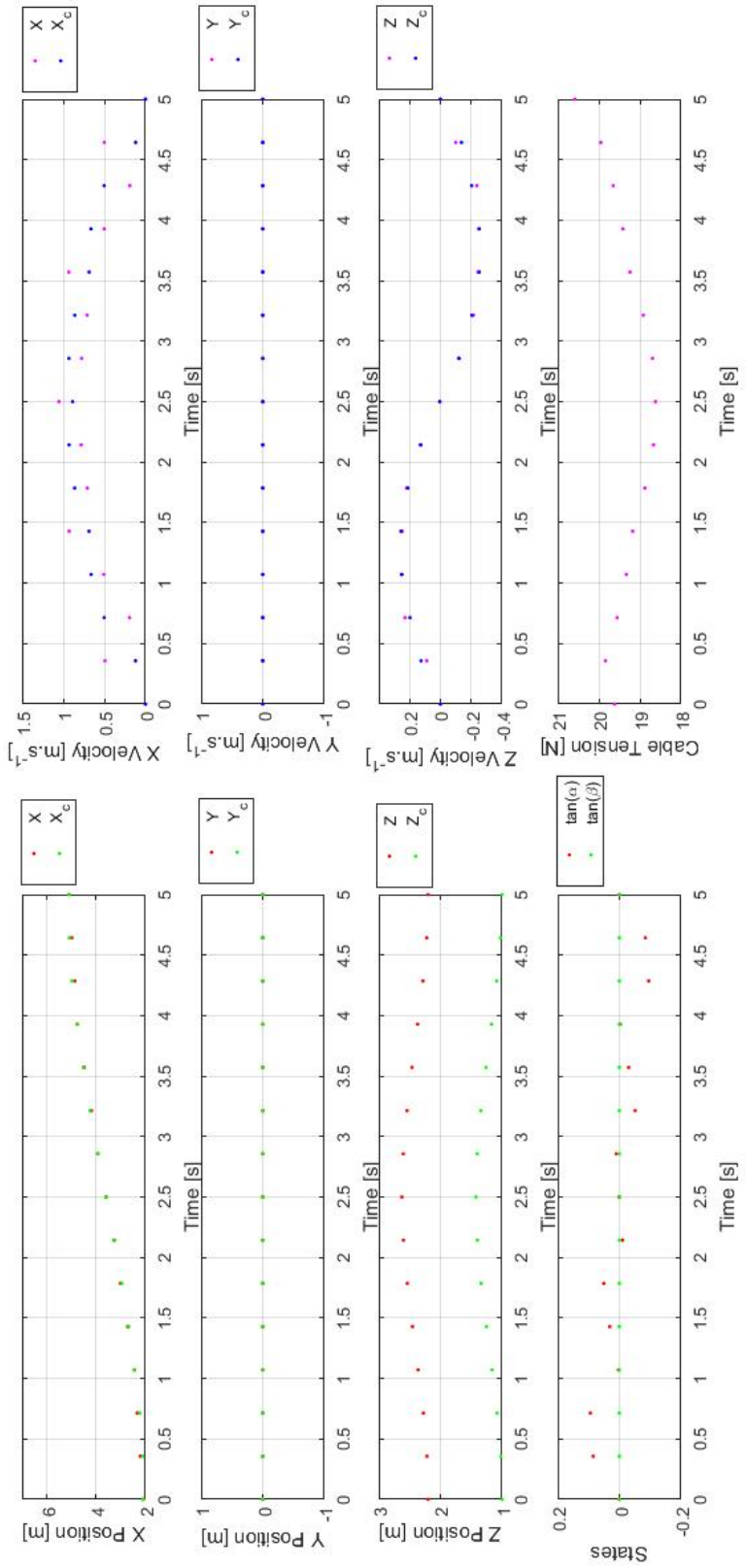


Figure 5.14: Evolution of System States (Single Agent with Payload, Open Loop)

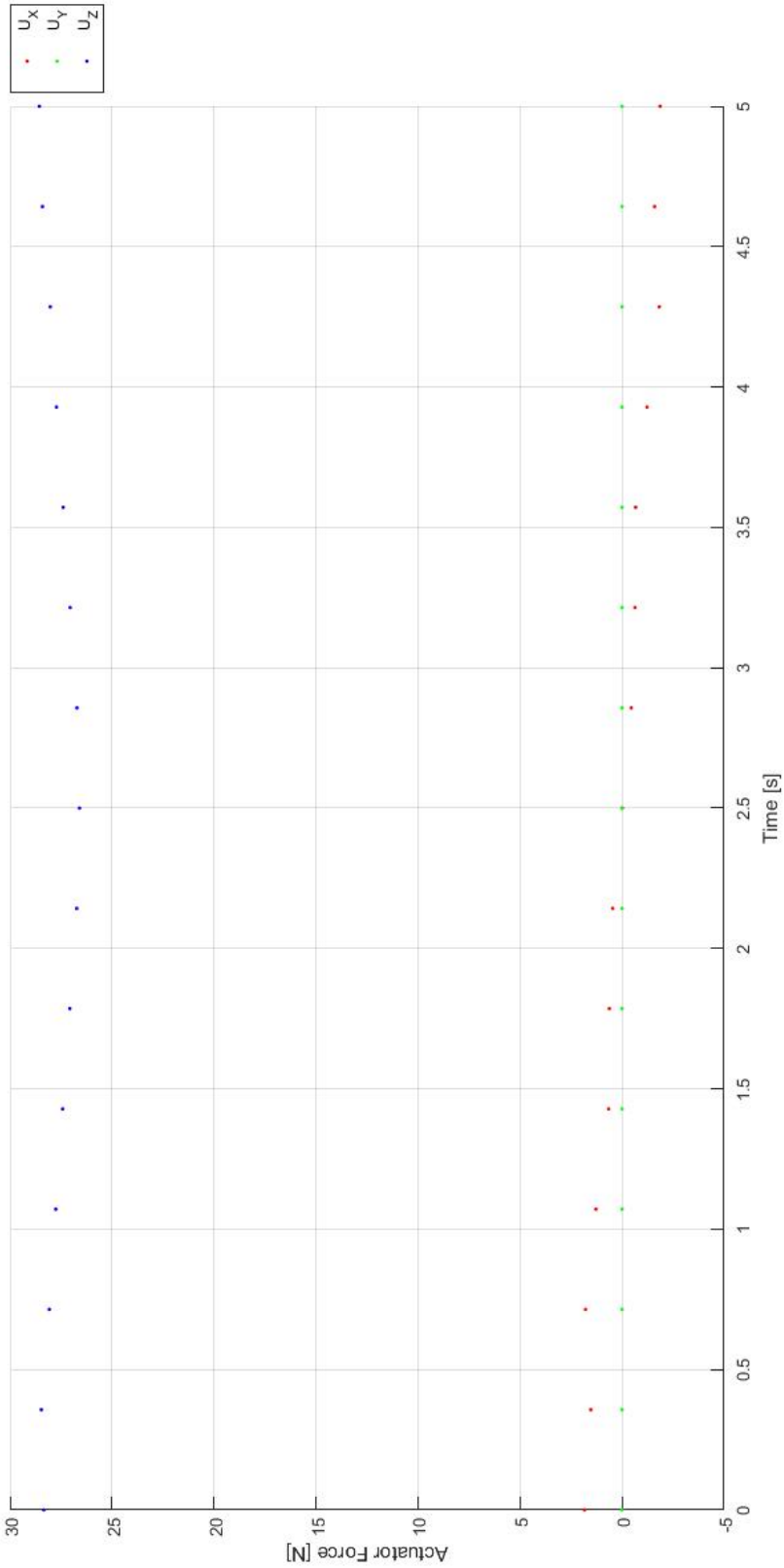


Figure 5.15: Evolution of System Inputs (Single Agent with Payload, Open Loop)

5.2.2 Closed-Loop Simulations with the ADM and GDM Models

Implementation: Next we have attempted to implement the MPC-based algorithm described in Section 4.2, using either the ADM model or the GDM model of the system in closed-loop with the controller. Using the same solver parameters as for the open-loop simulation, the closed-loop simulation using the ADM model exhibits positive feedback in the evolution of the system states (Figure 5.16) when the system is fed the (open-loop) optimal sequence of inputs at the beginning of the simulation (Figure 5.18). Due to this phenomenon we have been unable to run the closed-loop simulations with the ADM model without violating the state constraints, so it is not possible to compare the computational runtime with that of the open-loop simulation.

Unlike with the ADM model, the closed-loop simulation with the GDM model does not exhibit positive feedback. For this reason ICLOCS2 can successfully reoptimize the control inputs every second, yielding the evolutions of the system states (Figure 5.17) and the input sequence (Figure 5.19). The computational runtime (1 minute, 45 seconds) is several orders of magnitude slower than that of the open-loop simulation (1.99 s) due to overhead delays in Simulink, and an order of magnitude slower than the closed-loop simulation with a GDM model of the single agent (17.4 s) due to the increased computational complexity of the DHOCP and GDM model dynamics.

Observations: We first consider the closed-loop simulation using the ADM model. From analysis of the linearization of the system's dynamics, we note that the linearization's state-to-state transition matrix has all of its eigenvalues on the imaginary axis, so we can conclude that the system is not asymptotically stable in open-loop (see Appendix A for the derivation). We can therefore expect that the presence of noise introduced by numerical integration in Simulink is sufficient to cause positive feedback in such a system. This phenomenon is clearly demonstrated in Figure 5.16, in which the x-axis position and velocity for the agent and payload quickly grow unbounded from the beginning of the mission. Consequently the tangent of the cable angle α diverges to infinity, equivalent to the cable angle converging to 90° (i.e. the cable is fully horizontal).

Considering the GDM model, in Figures 5.17 and 5.19 we observe that the closed-loop evolutions of the system states and input sequence are consistent with those of the open-loop simulation (Figures 5.14 and 5.15 respectively), with a greater number of mesh points and a piecewise constant representation of the inputs $\{u_x, u_y, u_z\}$ and the cable tension F. Due to the increased number of mesh points in the closed-loop simulation, we can observe high-frequency oscillations in the cable tension F resembling a pair of lines in Figure 5.17, which are not captured by the open-loop simulation.

Mission Visualization: A series of still images from the mission visualization of the single-agent system with a payload are depicted in Figure 5.20. In order to make the agent's height changes more obvious we have superimposed a line on each image representing the initial height.

Consistent with the evolution of the agent's x and z positions in Figure 5.17, the agent progresses in the $+x$ direction between Figures 5.20a–5.20f. The agent's acceleration and subsequent deceleration in the x-axis reflect the evolution of the agent's thrust u_x in Figure 5.19. During the mission the agent alternates between leading the payload and lagging behind it: even when the agent is accelerating in the first half of the mission, Figure 5.20b depicts an instant where the agent lags behind the payload. Finally, we can observe an increase and decrease in the agent's z position and z velocity that are consistent with obstacle avoidance in Figure 5.17.

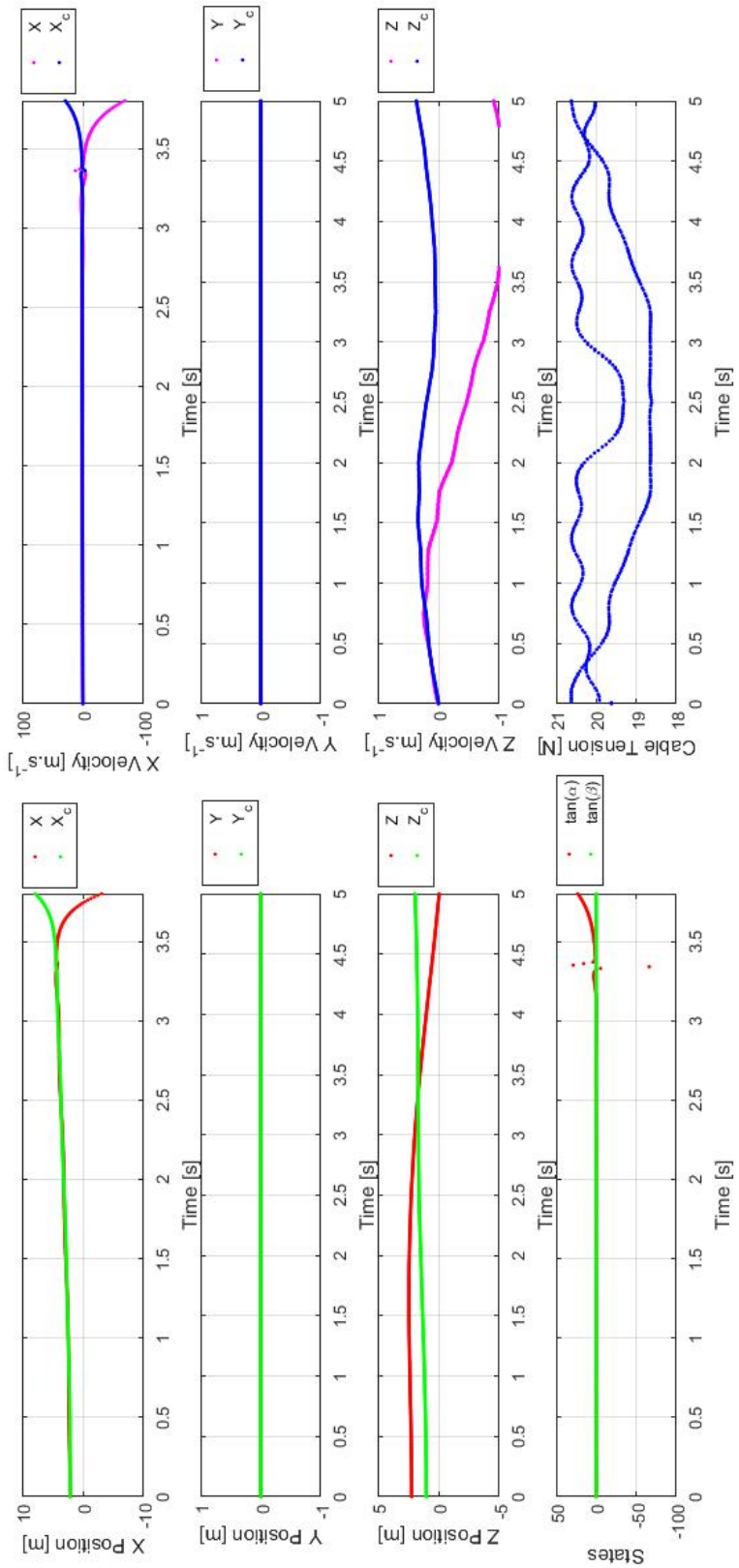


Figure 5.16: Evolution of System States (Single Agent with Payload, Closed Loop with ADM Model)

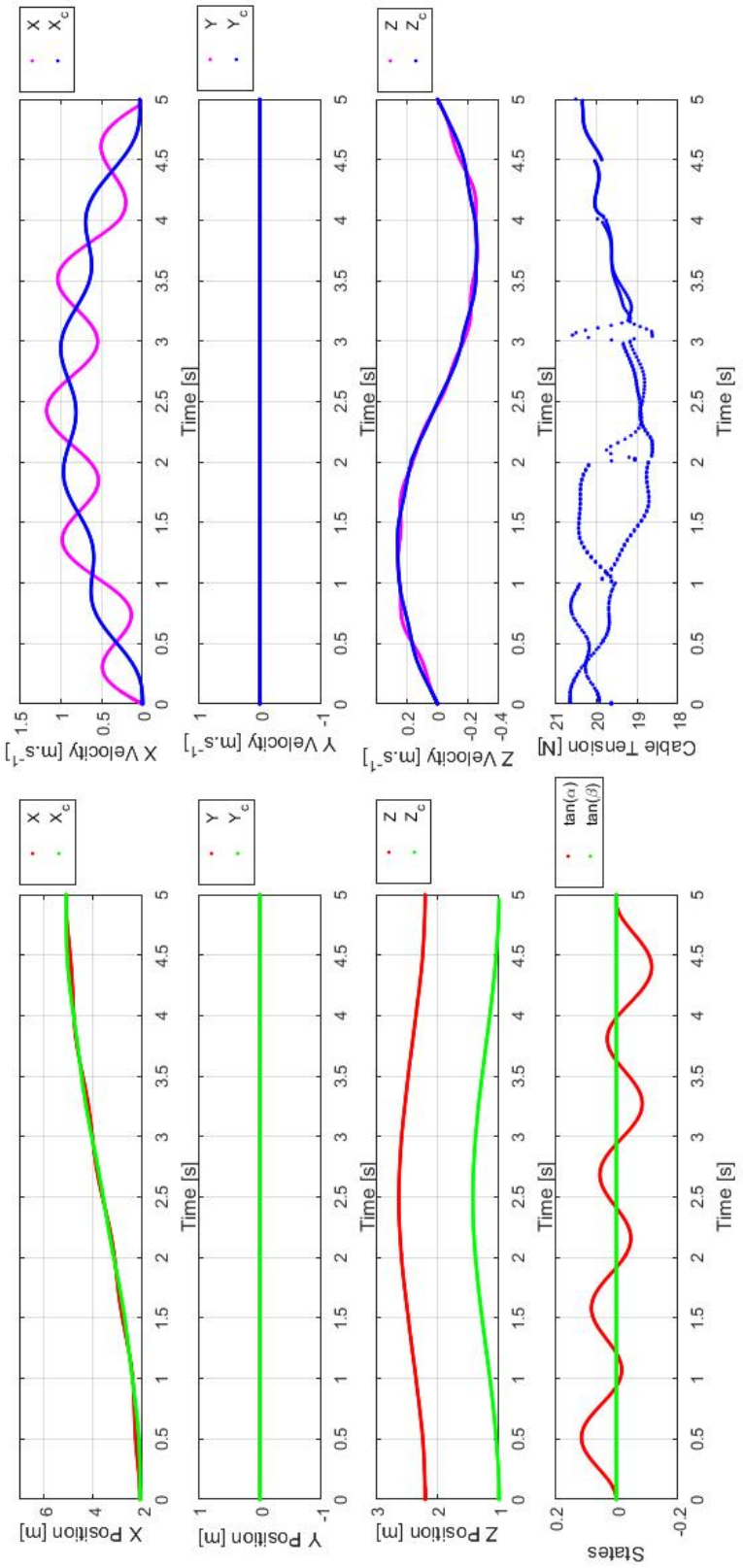


Figure 5.17: Evolution of System States (Single Agent with Payload, Closed Loop with GDM Model)

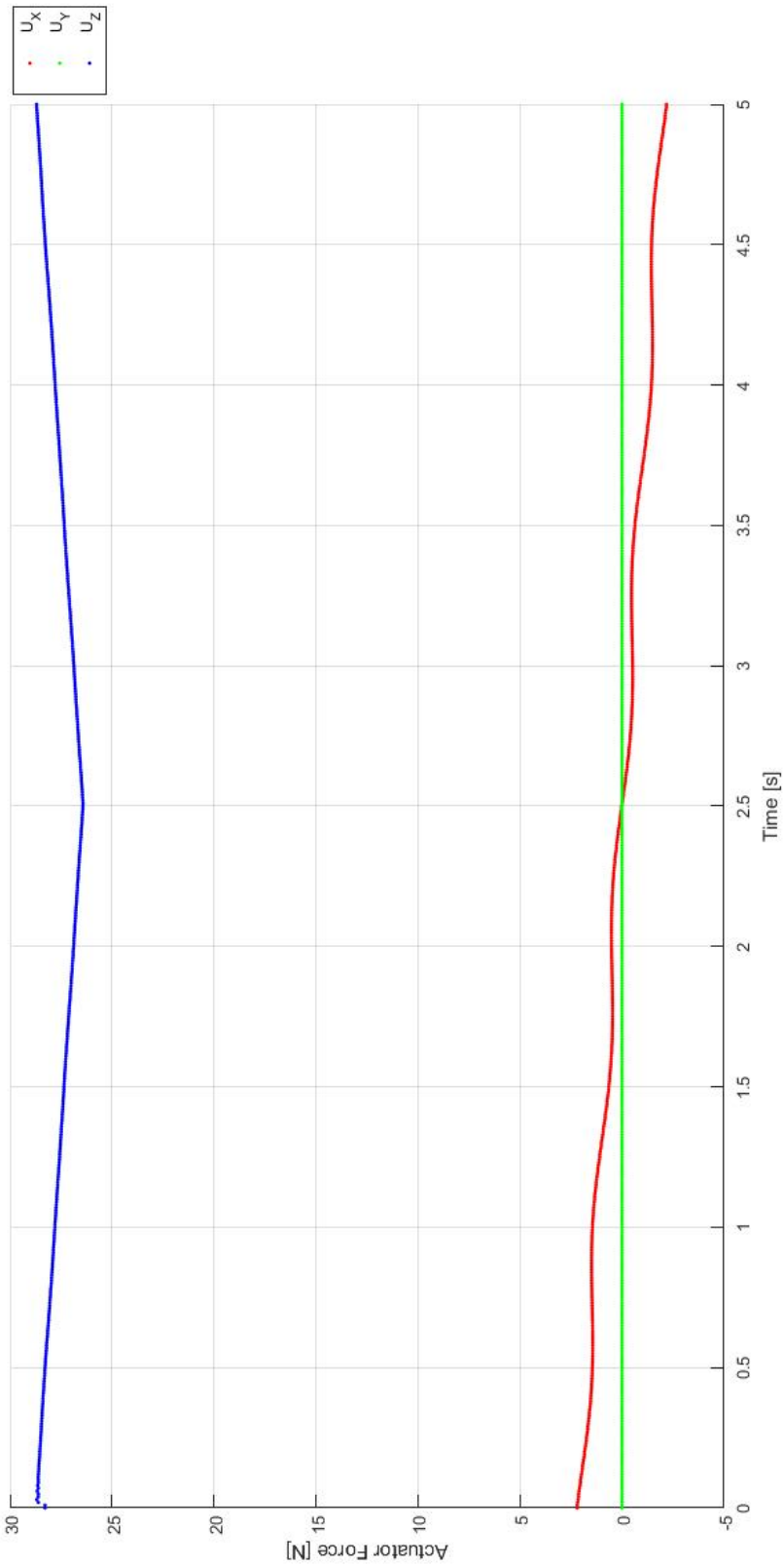


Figure 5.18: Evolution of System Inputs (Single Agent with Payload, Closed Loop with ADM Model)

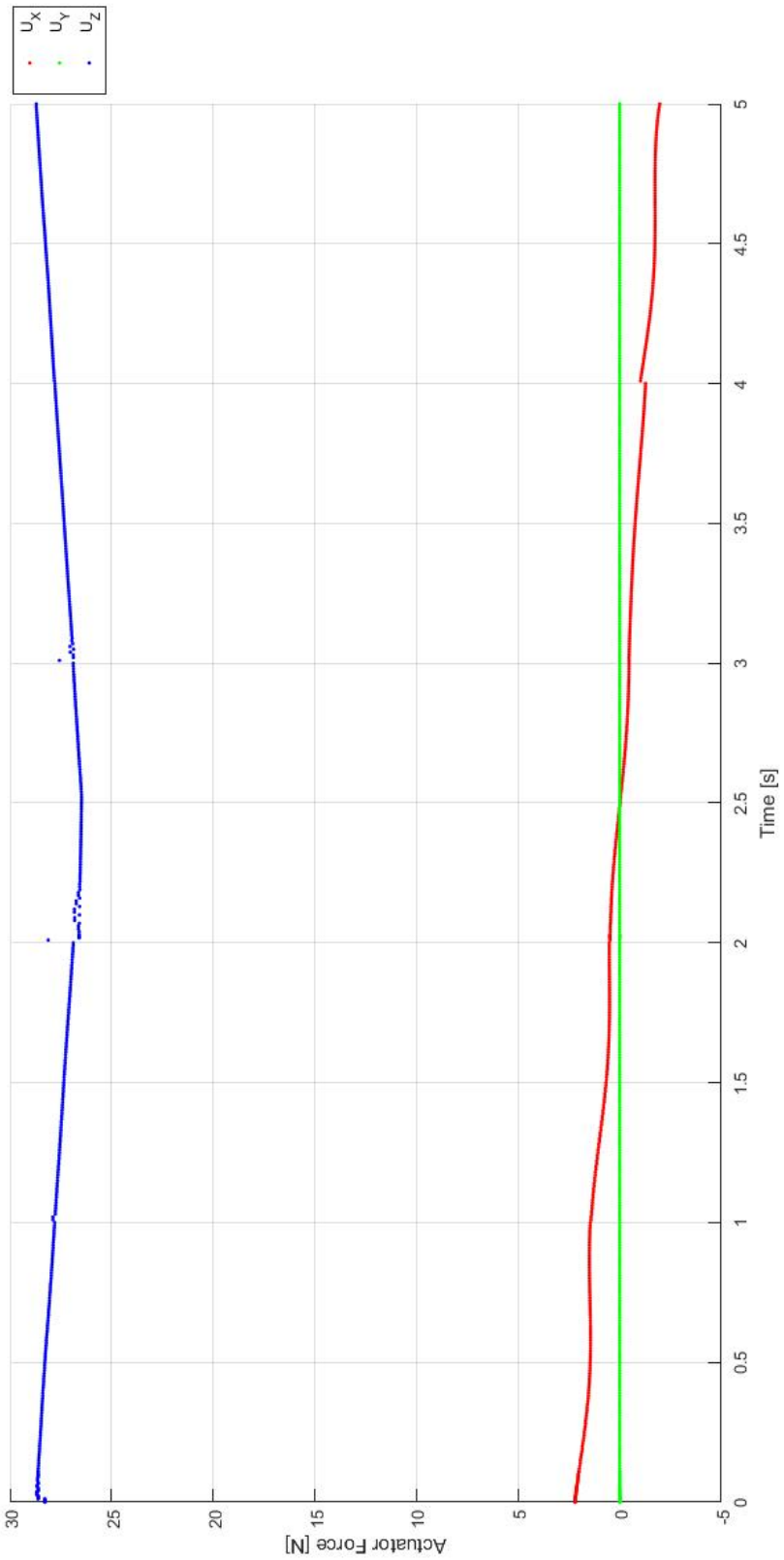
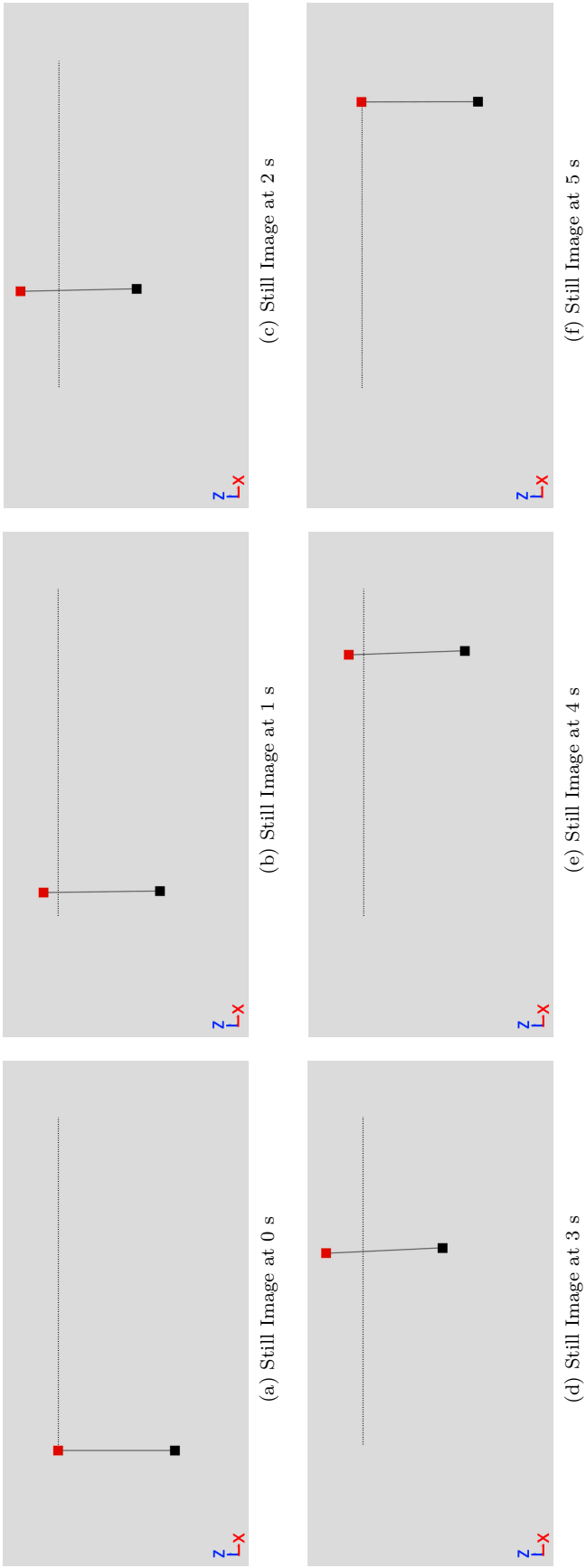


Figure 5.19: Evolution of System Inputs (Single Agent with Payload, Closed Loop with GDM Model)



(a) Still Image at 0 s (b) Still Image at 1 s (c) Still Image at 2 s

(d) Still Image at 3 s (e) Still Image at 4 s (f) Still Image at 5 s

Figure 5.20: Mission Visualization (Single Agent with Payload, Closed Loop with GDM Model)

5.3 Centralized Dual-Agent CPT

We now proceed to construct a dual-agent system with a cable-suspended payload from two instances of a single agent with a cable-suspended payload.

5.3.1 Open-Loop Simulations

Error Analysis: We have conducted a series of performance tests using combinations of the three transcription methods (the Euler, Trapezoidal, and Hermite-Simpson methods) and various numbers of mesh points {10,20,40,60,100}. We have performed the open-loop simulations for each combination and extracted the computation time (Figure 5.21) and maximum error (Figure 5.22).

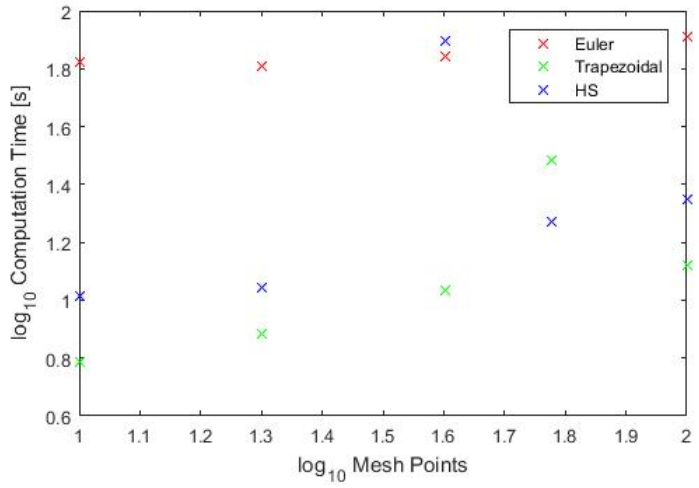


Figure 5.21: Computation Time (Two Agents with Payload, Open Loop)

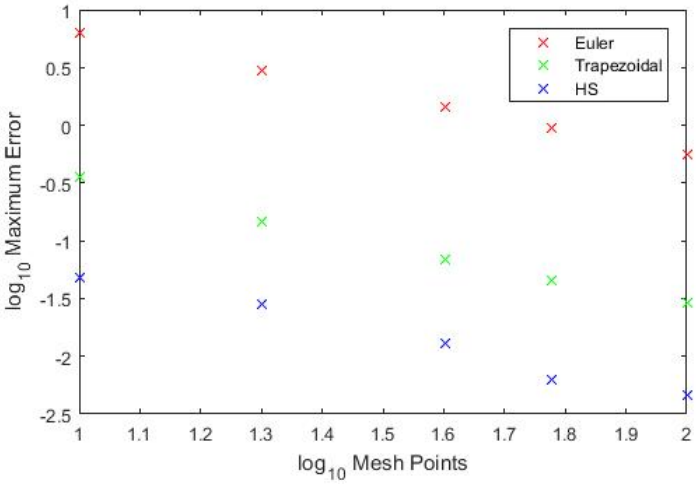


Figure 5.22: Maximum Error (Two Agents with Payload, Open Loop)

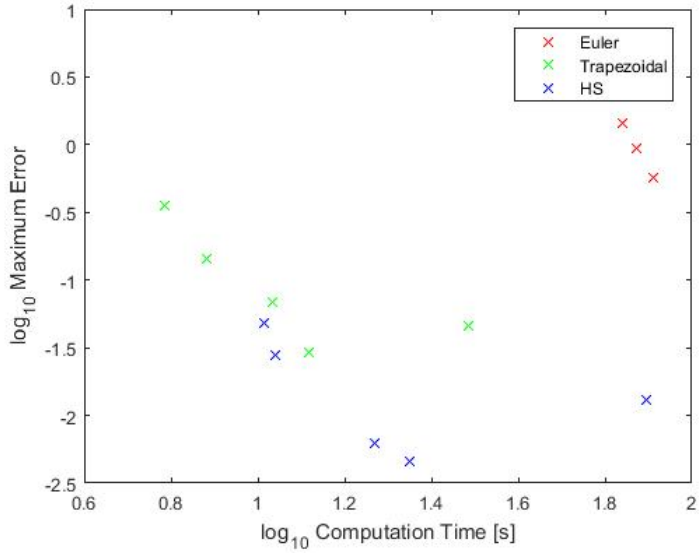


Figure 5.23: Maximum Error v. Computation Time (Two Agents with Payload, Open Loop)

Plotting the computation time against maximum error yields the graph in Figure 5.23. In these three graphs we notice the same phenomena as in Section 5.1.1, with the computation times in Figure 5.21 longer than those of the single-agent system with a payload due to the increased complexity of the DHOCP. To avoid waiting for an extended period to find an optimal solution, we have imposed an upper limit of 500 iterations on the solver.

Solver Parameter Selection: We have selected the combination of transcription method and number of mesh points that yields an optimal solution with the fastest computation time and a suitably low error. A Hermite-Simpson transcription method with 10 mesh points runs in 10.7 s, an order of magnitude slower than the open-loop simulation of a single agent with a payload (1.99 s). We have then used these parameters to solve the DHOCP described in Section 4.3, yielding the evolutions of the system states (Figure 5.24) and the input sequence (Figure 5.25). Due to the fact that the dual-agent model is a constrained duplication of the single-agent system with a cable-suspended payload, we can observe that the agents have identical inputs. At all times the computed solution complies with all constraints on the state and input values.

Physical Interpretation: In each axis of motion the dual-agent system's behavior is qualitatively similar to that of the single agent with a cable-suspended payload.

Initially the i^{th} agent's thrust in the z-axis is increased above the hovering thrust (equal to $(m_{ag}+\frac{m_{pl}}{2})g = 1.81 * 9.81 = 17.8\,\text{N}$) to cause a net upward acceleration. Likewise the cable tensions F_i initially exceed the hovering thrust for the payload (equal to $\frac{m_{pl}}{2}g = 1{*}9.81 = 9.81\,\text{N}$) to cause a net upward acceleration. Note that the hovering thrusts for each agent and each payload are reduced from those of the single-agent system with a payload due to the shared distribution of the payload's mass between two agents. Both u_{zi} and F_i steadily decrease until the agent reaches the peak of the obstacle, at which the z velocities of the agent and payload changes from positive to negative and both enter freefall. Both u_{zi} and F_i then increase to counteract freefall, bringing the agent and the payload to their respective target heights at the end of the mission.

Thrust in the x-axis is initially positive, causing the i^{th} agent to accelerate. The thrust remains constant for approximately 1.25 s to allow the agent and the payload to accelerate. We notice that the payload initially lags behind the agent, hence $\tan\alpha_i$ is positive. The thrust begins to decrease after 1.25 s, causing the agent to slow down and the payload to temporarily lead the agent. The thrust crosses from positive to negative as soon as the agent reaches the midway point, causing the x velocities of the agent and

payload to decrease. In this part of the mission the payload tends to lead the agent, hence $\tan \alpha_i$ is negative. At the end of the mission the agent briefly accelerates to counteract any overshoot from the payload, allowing both the agent and the payload to stop at the target position. It is important to observe that the oscillations in $\tan \alpha_i$ are slower than those of the single-agent system with a payload, suggesting the presence of damping in the mechanical configuration of the dual-agent system.

Thrust in the y-axis is negligible as desired; the non-zero input values can be attributed to numerical errors from the optimization solver.

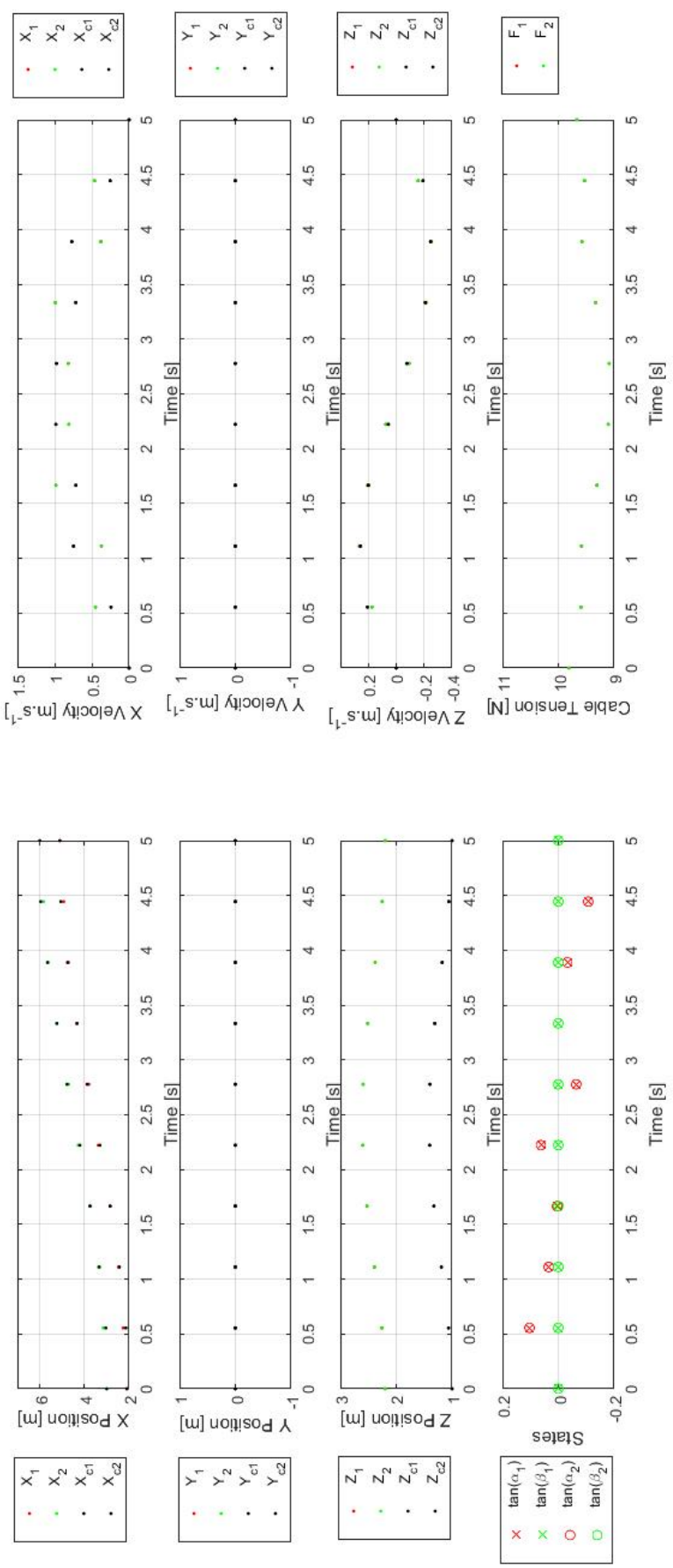


Figure 5.24: Evolution of System States (Two Agents with Payload, Open Loop)

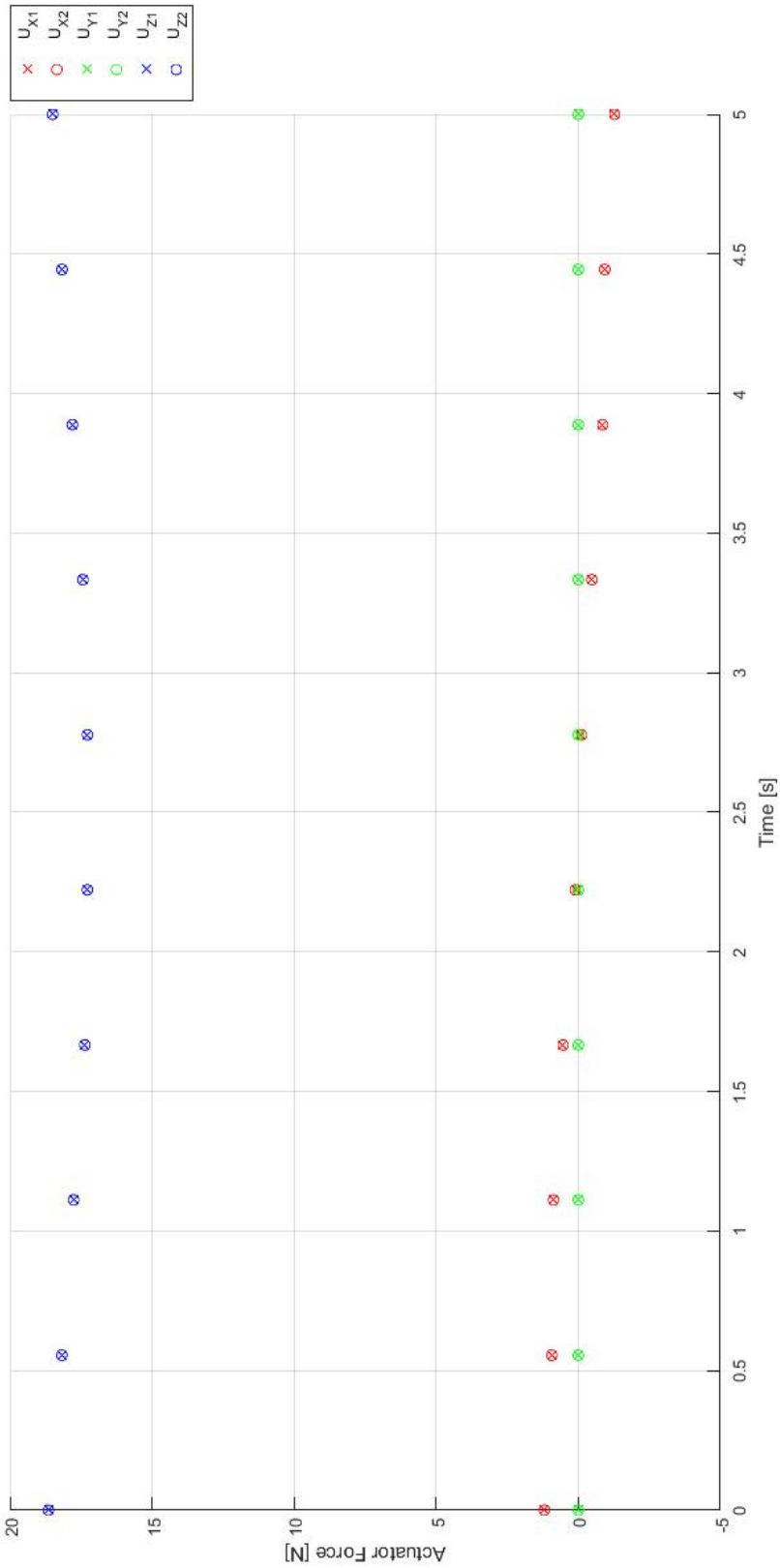


Figure 5.25: Evolution of System Inputs (Two Agents with Payload, Open Loop)

5.3.2 Closed-Loop Simulations with the GDM Model

Implementation: Due to the runtime issues encountered with the ADM model in Section 5.2.2, we have focused on closed-loop simulations with the GDM model only. Using the same solver parameters as in Section 5.3.1 with reoptimization every second, ICLOCS2 has produced the closed-loop evolutions of the system states (Figure 5.26) and input sequence (Figure 5.27).

Observations: The closed-loop evolutions of the system states and input sequence resemble interpolated versions of those of the open-loop simulation (Figures 5.24 and 5.25 respectively). Unlike the open-loop simulation, there are discontinuities in the closed-loop inputs for the x- and z-axes at moments of reoptimization in Figure 5.27. Due to the increased number of mesh points in the closed-loop simulation, in Figure 5.26 we observe high-frequency oscillations in the cable tensions $\{F_1, F_2\}$ (resembling a pair of lines) which are not captured by the open-loop simulation. Finally, the computational runtime (8 minutes, 57 seconds) is several orders of magnitude slower than that of the open-loop simulation (10.7 s) due to overhead delays in Simulink, and an order of magnitude slower than that of the closed-loop simulation of the single-agent system with a payload (1 minute, 45 seconds) due to the increased complexity of the DHOCP and the GDM model dynamics.

Mission Visualization: A series of still images from the mission visualization of the dual-agent system are depicted in Figure 5.28. In order to make the agents' height changes more obvious we have superimposed a line on each image representing the agents' initial height.

Consistent with the evolution of the agent's x and z positions in open-loop (Figure 5.24) and closed-loop (Figure 5.26), the agents move in the $+x$ direction between Figures 5.28a–5.28f. The agents' intermittent acceleration and deceleration in the x-axis reflect the changes in the agents' x inputs in Figure 5.27. As per the oscillations in $\tan\alpha_i$ in Figure 5.26, during the mission the agent alternates between leading the payload and lagging behind it, with the swing of the payload masses less pronounced than in the visualization of the single agent with a payload (Figure 5.20). We also observe an increase and decrease in the agents' z positions and z velocities that are consistent with obstacle avoidance in Figures 5.24 and 5.26.

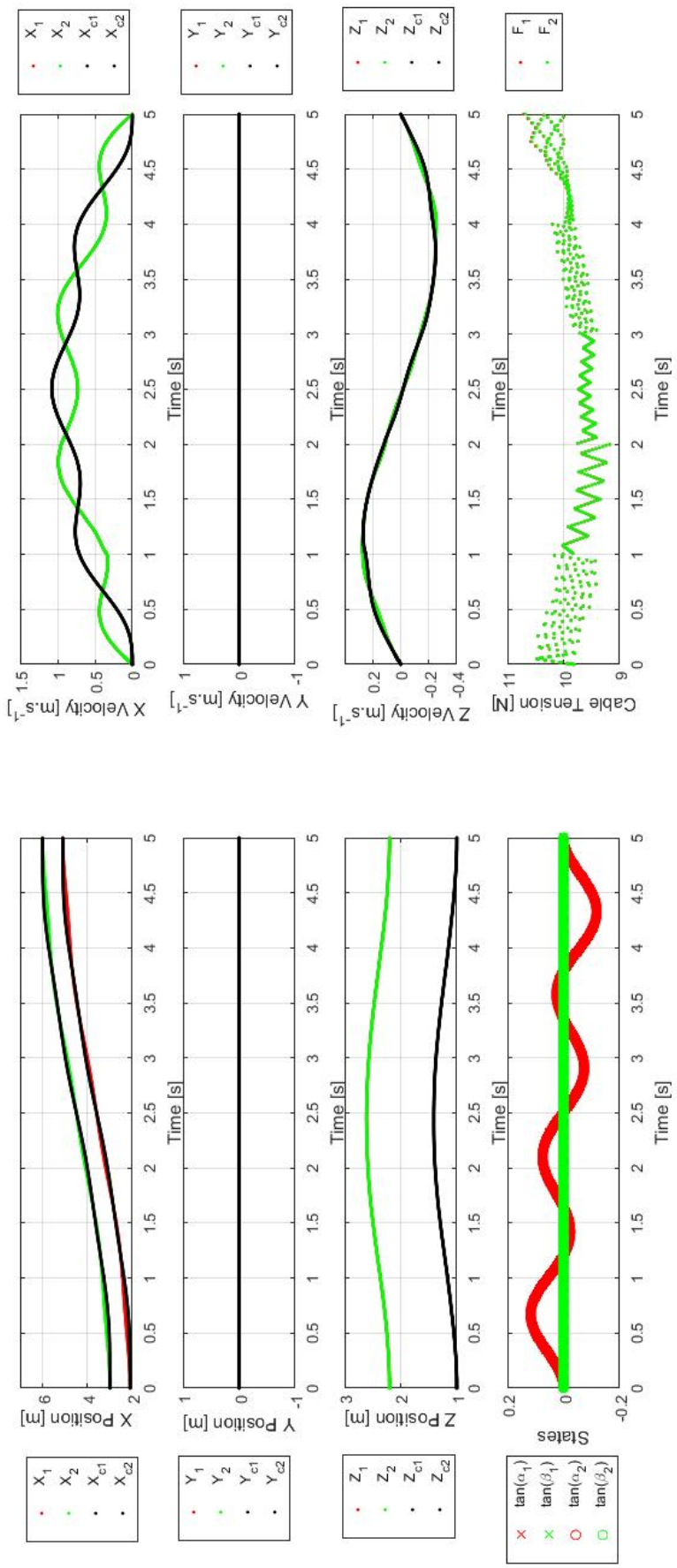


Figure 5.26: Evolution of System States (Two Agents with Payload, Closed Loop with GDM Model)

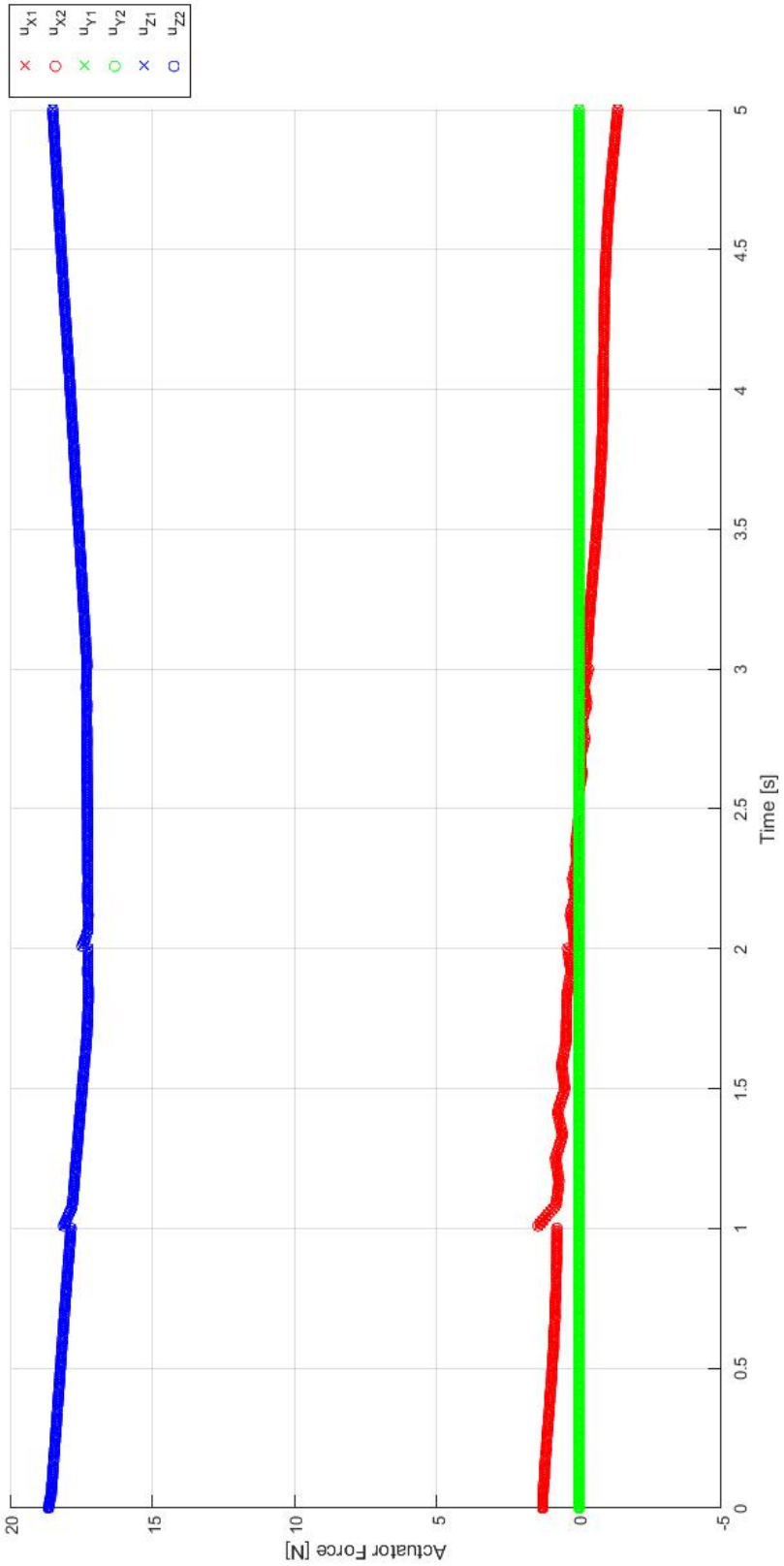
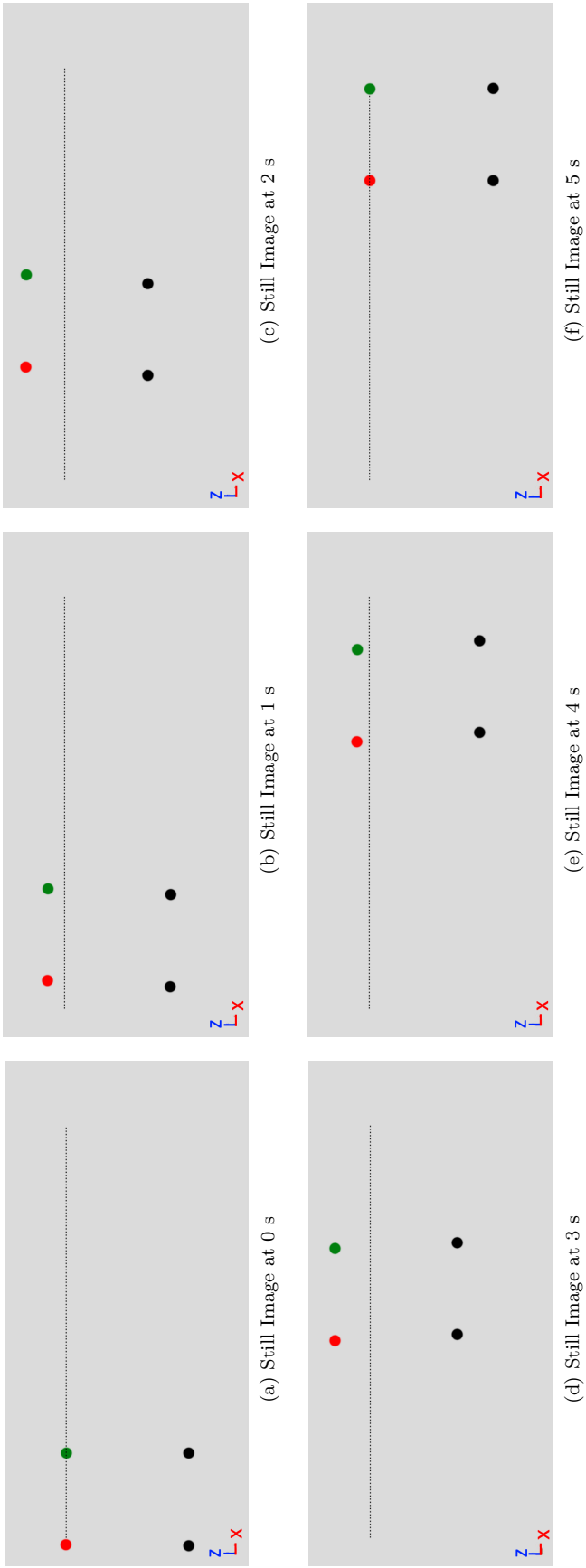


Figure 5.27: Evolution of System Inputs (Two Agents with Payload, Closed Loop with GDM Model)



(a) Still Image at 0 s (b) Still Image at 1 s (c) Still Image at 2 s

(d) Still Image at 3 s (e) Still Image at 4 s (f) Still Image at 5 s

Figure 5.28: Mission Visualization (Two Agents with Payload, Closed Loop with GDM Model)

5.4 Centralized Triple-Agent CPT

In this section we construct a triple-agent system with a cable-suspended payload from three instances of a single agent with a cable-suspended payload.

Experimentally we have found that the solver cannot find a feasible solution for the triple-agent CPT problem as defined in (3.30), hence we have redefined the DHOCP as follows:

$$\min_{U(t),\, s} \left\| \overline{X}(t_k) - \overline{X}(t_f) \right\|^2 + \sum_{\tau=t_k}^{t_f} \|U(\tau)\|^2 + |s| \tag{5.1a}$$

subject to

$$\sqrt{(x_{ci}(t) - x_i(t))^2 + (y_{ci}(t) - y_i(t))^2 + (z_{ci}(t) - z_i(t))^2} = L,\ \forall i \in \{1, ..., N\},\ \forall t \in [0, t_f] \tag{5.1b}$$

$$\sqrt{(x_i(t) - x_j(t))^2 + (y_i(t) - y_j(t))^2 + (z_i(t) - z_j(t))^2} > \sigma,\ \forall i, j \in \{1, ..., N\},\ i \neq j,\ \forall t \in [0, t_f] \tag{5.1c}$$

$$z_{c2}(t) - h(\tanh(x_{c2}(t) - x_a) - \tanh(x_{c2}(t) - x_b)) - \kappa - s = 0,\ \kappa > 0,\ s > 0,\ \forall t \in [0, t_f] \tag{5.1d}$$

where the cable length L=1.20 m, the separation constant σ=0.90 m, the obstacle's height h=1.00 m, the clearance κ=0.50 m, and the parameters x_a=4.10 m and x_b=4.90 m correspond respectively to the start and end of the obstacle's length in the x-axis.

In (5.1d) we have imposed one obstacle path constraint defined with respect to the position of the middle payload mass. In addition, we have included a slack variable s in the single obstacle path constraint. Minimizing this slack variable allows the solver to find a solution with the smallest possible violation of a 'hard' path constraint if no feasible solution can be found [39]. We have chosen to weight the slack variable in the cost function (5.1a) using the L_1 norm instead of the L_2 norm, since the former yields a marginally faster runtime in open-loop simulations (67.1 s as opposed to 70.8 s).

5.4.1 Open-Loop Simulations

Error Analysis: As in the previous section, we have conducted a series of performance tests using combinations of three transcription methods (the Euler, Trapezoidal, and Hermite-Simpson methods) and various numbers of mesh points {10,20,40,60,100}. We have performed the open-loop simulations for each combination and extracted the computation time (Figure 5.29) and maximum error (Figure 5.30).

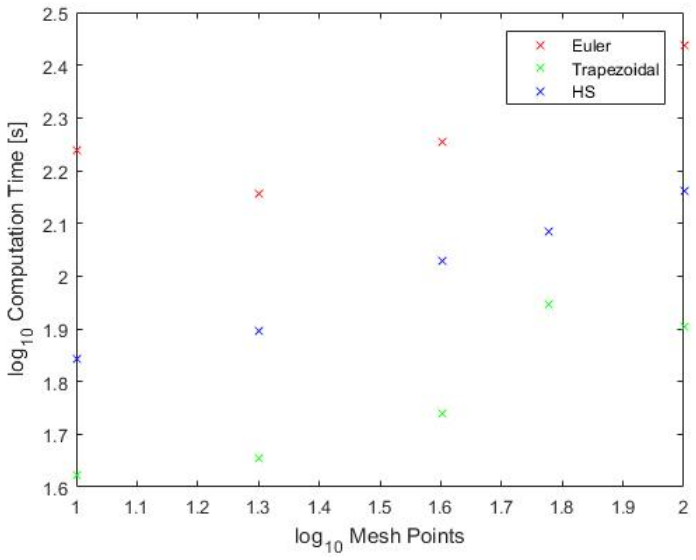


Figure 5.29: Computation Time (Three Agents with Payload, Open Loop)

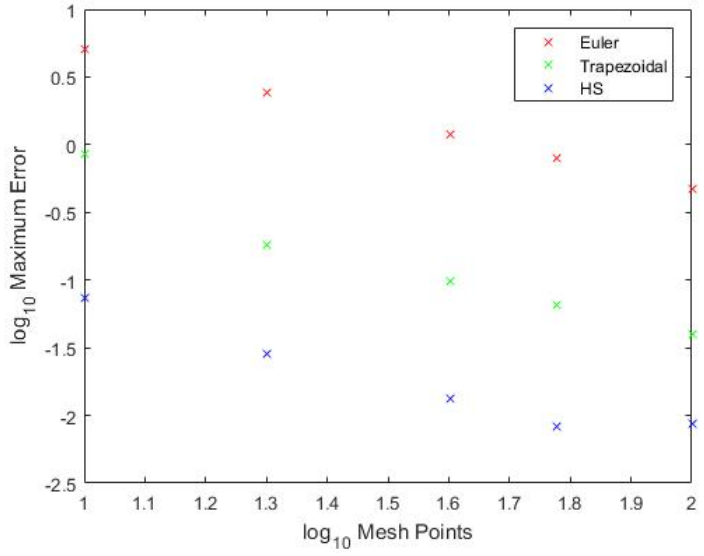


Figure 5.30: Maximum Error (Three Agents with Payload, Open Loop)

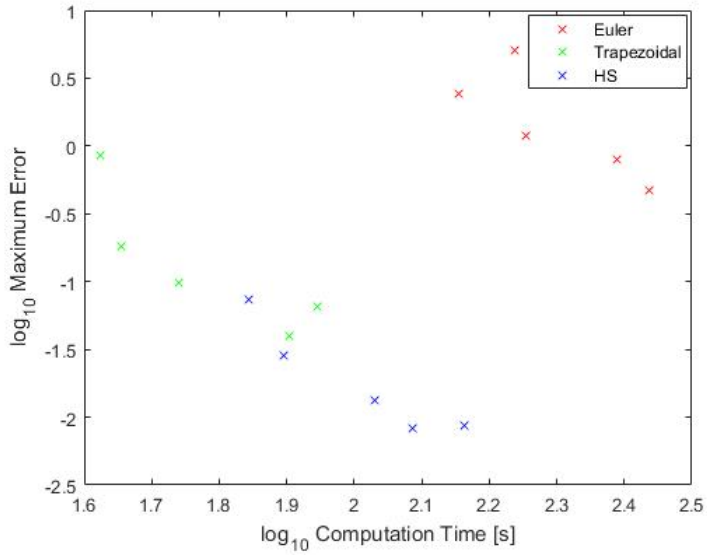


Figure 5.31: Maximum Error v. Computation Time (Three Agents with Payload, Open Loop)

Plotting the computation time against maximum error yields the graph in Figure 5.31. We again notice the same phenomena as in Section 5.1.1. The computation times in Figure 5.29 are much longer than those of the single agent with a payload, due to the increased number of states (36) and of nonlinear path constraints associated with the payload rigidity and cable lengths.

Solver Parameter Selection: We have selected the combination of transcription method and number of mesh points that yields an optimal solution with a fast computation time and a suitably low error. A Hermite-Simpson transcription method with 10 mesh points runs in 67.1 s. We have then used these parameters to solve the DHOCP described in Section 5.4 with obstacle avoidance, yielding the evolutions of the system states (Figure 5.32) and the input sequence (Figure 5.33). At all times the computed solution complies with all constraints on the state and input values.

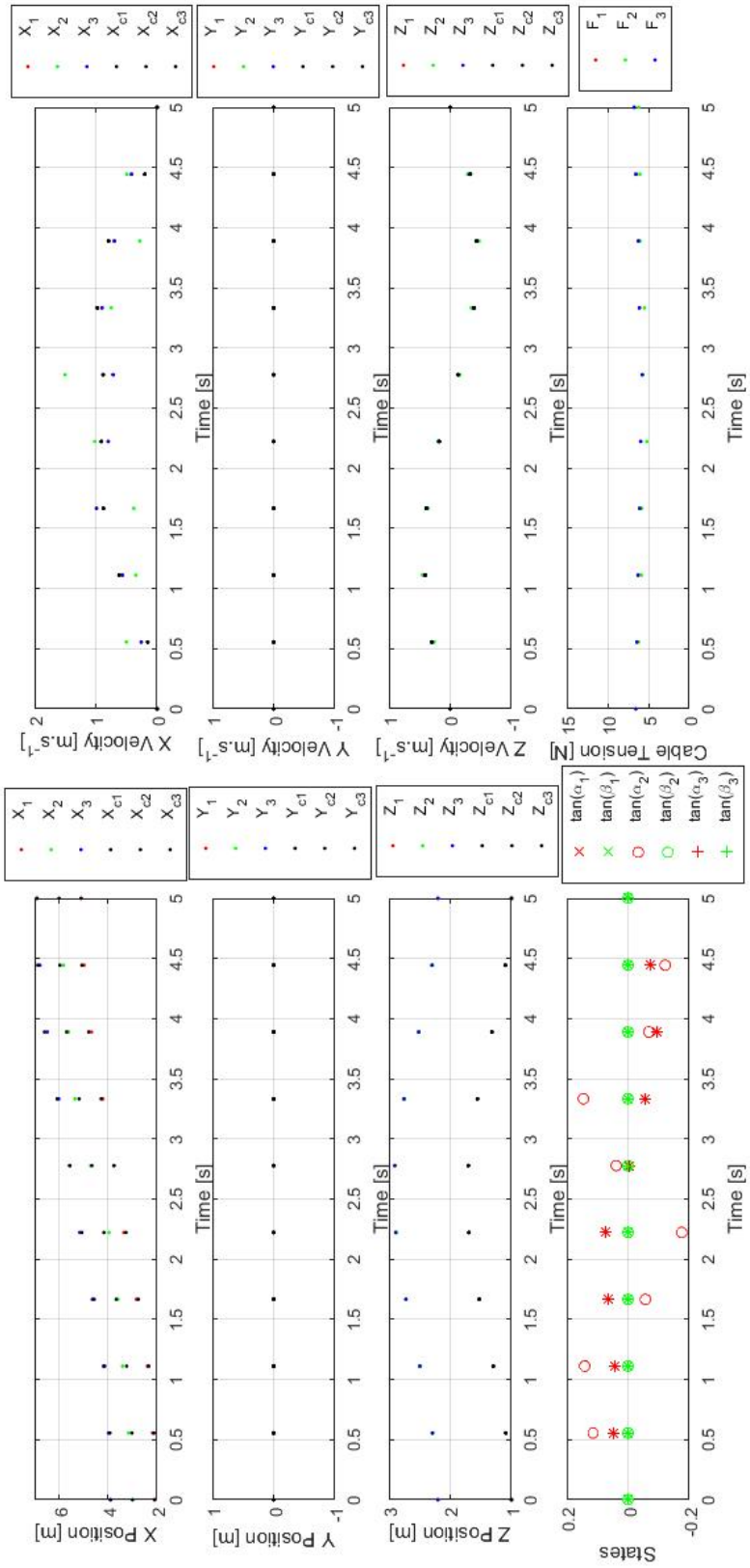


Figure 5.32: Evolution of System States (Three Agents with Payload, Open Loop)

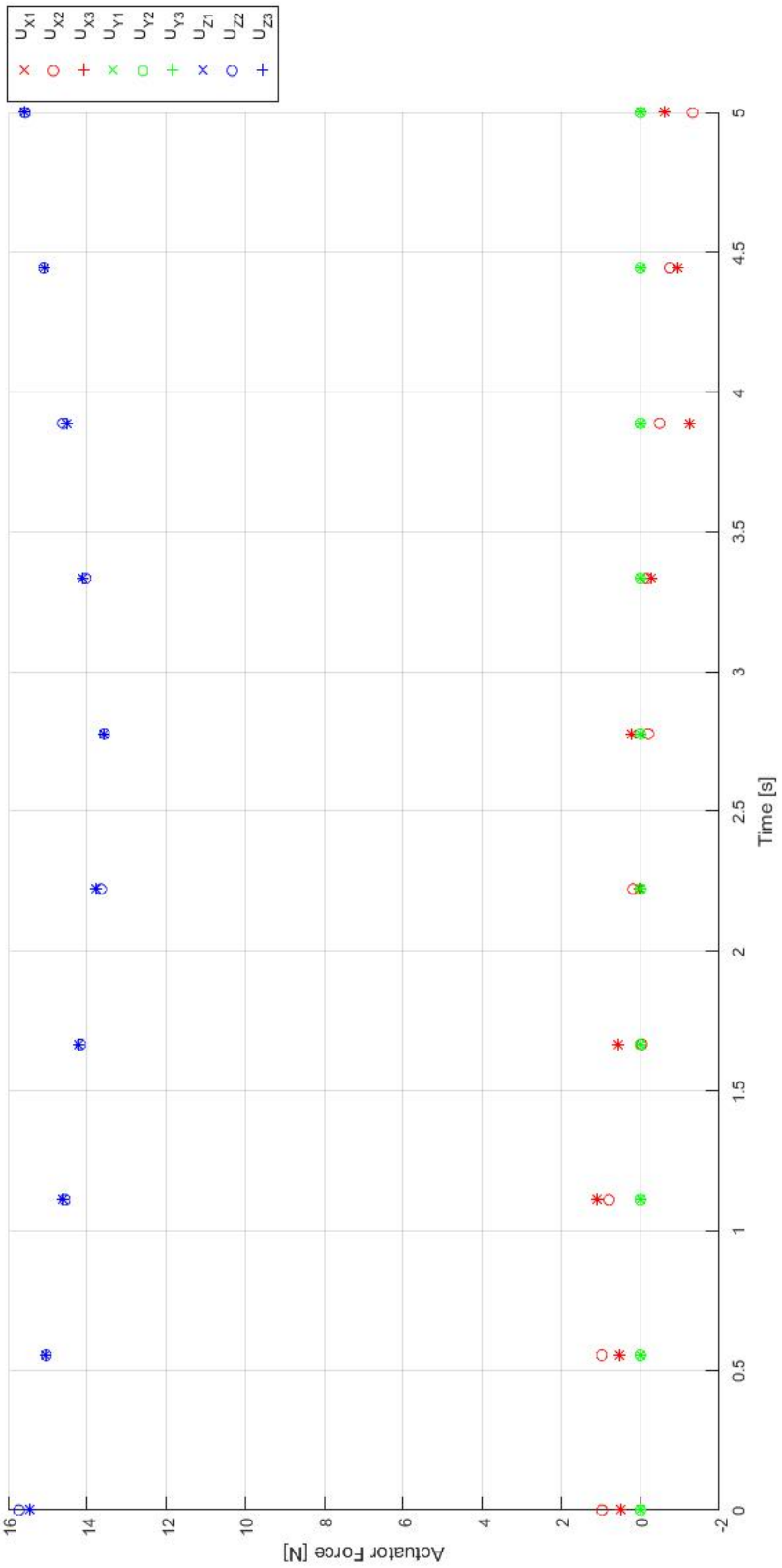


Figure 5.33: Evolution of System Inputs (Three Agents with Payload, Open Loop)

Physical Interpretation: In Figure 5.33 we notice that the system navigates in the presence of a spatial obstacle in the X-Z plane by varying the thrust of each agent in the x- and z-axes. Since the triple-agent model is a constrained combination of three individual agents with cable-suspended payloads, the analysis is qualitatively similar to that of the dual-agent CPT system, however the second agent has different input values to those of the other agents. Since the obstacle path constraint is imposed directly on the second agent-payload system, the second agent must first adjust its position in the x- and z-axes so that the payload avoids the obstacle, and the other agents must move to comply with the path constraints on cable length and payload rigidity.

Initially the i^{th} agent's thrust in the z-axis u_{zi} is increased above the hovering thrust (equal to $(m_{ag} + \frac{m_{pl}}{3})g = 1.48 * 9.81 = 14.5\,\text{N}$) to cause a net upward acceleration, with u_{z2} slightly greater than u_{z1} and u_{z3}. Likewise the cable tensions F_i initially exceed the hovering thrust required for the payload (equal to $\frac{m_{pl}}{3}g = \frac{2}{3} * 9.81 = 6.54\,\text{N}$) to cause a net upward acceleration. Note that the hovering thrusts for each agent and payload attachment point are reduced from those of the dual-agent CPT system due to the shared distribution of the payload's mass between three agents. Both u_{zi} and F_i steadily decrease until the middle agent reaches the peak of the obstacle, at which u_{z2} is less than u_{z1} and u_{z3}. The z velocities of each agent and attachment point changes from positive to negative, so all enter freefall. Both u_{zi} and F_i then increase to counteract freefall, bringing the agents and the payload to their respective target heights at the end of the mission.

As for the dual-agent CPT system, thrust in the x-axis u_{xi} is initially positive with u_{x2} greater than u_{x1} and u_{x3}, so the second agent accelerates faster than the other agents. Each agent's thrust remains constant for approximately 750 ms to allow the payload to accelerate, with the second agent accelerating faster than the other two agents. We notice that the payload's attachment points initially lag behind their respective agent so $\tan\alpha_i$ is positive. Reflecting the relative motion of the second agent with respect to the other agents, $\tan\alpha_2$ oscillates faster than $\tan\alpha_1$ and $\tan\alpha_3$.

After 750 ms u_{x2} begins to decrease, causing the agent to slow down and the payload's second attachment point to temporarily lead the agent. At the same time u_{x1} and u_{x3} briefly rise, so that the first and third agents can maintain their separation from the second agent. When the second agent reaches the midway point, u_{x2} crosses from positive to negative, causing the x velocity of the second attachment point to decrease. Near 3 s u_{x1} and u_{x3} cross from positive to negative, so the first and third attachment points' velocities decrease. In this part of the mission the first and third attachment points lead the first and third agents so $\tan\alpha_1$ and $\tan\alpha_3$ are negative, whereas $\tan\alpha_2$ oscillates according to the position of the second agent relative to the second attachment point. At the end of the mission each agent undergoes negative acceleration to counteract any overshoot from the payload, allowing each agent and the payload to stop at their target positions.

Thrust in the y-axis u_{yi} is negligible, as desired. The non-zero input values can be attributed to numerical errors from the optimization solver.

5.4.2 Closed-Loop Simulations with the GDM Model

Implementation: Due to the runtime issues with the ADM model of a pendulum in Simulink mentioned in Section 5.2.2, we have completed closed-loop simulations with a GDM model only. To avoid waiting for an extended period to find a feasible solution for (5.1), we have imposed an upper limit of 40 iterations on the solver. Using the same simulation parameters as in Section 5.4.1 with reoptimization every second, ICLOCS2 has produced the closed-loop evolutions of the system states (Figure 5.34) and input sequence (Figure 5.35).

Observations: The closed-loop evolutions of the system states and input sequence resemble interpolated versions of those of the open-loop simulation (Figures 5.32 and 5.33 respectively). Unlike the previous open-loop simulation, each agent's thrusts are identical. Due to the increased number of mesh points in the closed-loop simulation, in Figure 5.34 the cable tensions $\{F_1, F_2, F_3\}$ exhibit high-frequency oscillations which are not captured by the open-loop simulation. We also note that F_2 differs from F_1 and F_3 in the last half of the mission. In the last second of the mission, F_2 appears to fluctuate perfectly out of phase with F_1 and F_3, suggesting moderate deformations in the payload as it approaches its terminal height. The cable tensions then converge to 10.8 N, exceeding the hovering thrust required for the payload (6.54 N). Given that the cable tensions' final value differs from that of the open-loop simulation in Figure 5.32, it is likely that in the final reoptimization at 4 s the solver converged to an erroneous evolution of the cable tensions and terminated without finding the optimal solution. The computational runtime (29 minutes, 26 seconds) is several orders of magnitude slower than that of the open-loop simulation (67.1 s) due to overhead delays in Simulink, and that of the closed-loop simulation of the single agent with a payload (1 minute, 45 seconds) due to the increased computational complexity of the DHOCP and the GDM model dynamics.

Mission Visualization: A series of still images from the mission visualization are depicted in Figure 5.36. To enable visual comparisons of height throughout the mission, we have superimposed a line on each image representing the agents' initial height. Consistent with the evolution of the agent's x and z positions in open-loop (Figure 5.32) and closed-loop (Figure 5.34), the agents move in the $+x$ direction between Figures 5.36a–5.36f. The agents' pattern of acceleration and deceleration in the x-axis reflect the changes in the agents' x inputs in Figures 5.33 and 5.35. As per Figure 5.34, each agent starts by leading its payload mass (so the cable angle is positive), then after 2 s each agent lags its payload mass as it slows down (so the cable angle is negative). We observe that in Figures 5.36a–5.36c each agent and its payload mass initially accelerate in the $+z$ direction, thus increasing their z positions to navigate over the spatial obstacle. Subsequently in Figures 5.36d–5.36f the z positions of each agent and its payload mass decrease until the target z positions are reached.

When compared with the dual-agent CPT system, the oscillations in $\tan\alpha_i$ have a similar period with reduced amplitude. Such behavior might be explained by the presence of damping inside the triple-agent configuration, reflecting cancellations of the most extreme differences in the forces exerted by each agent on the payload.

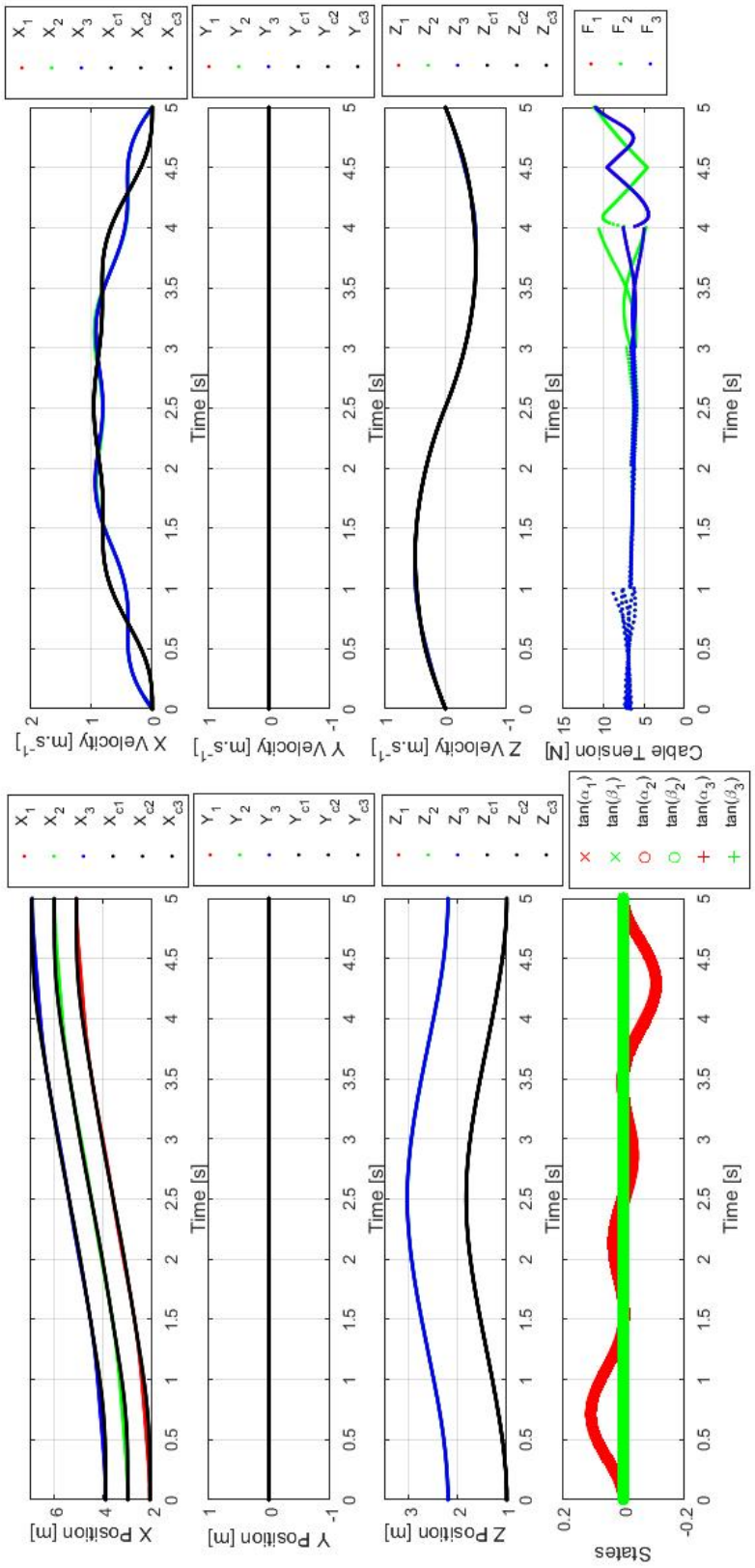


Figure 5.34: Evolution of System States (Three Agents with Payload, Closed Loop with GDM Model)

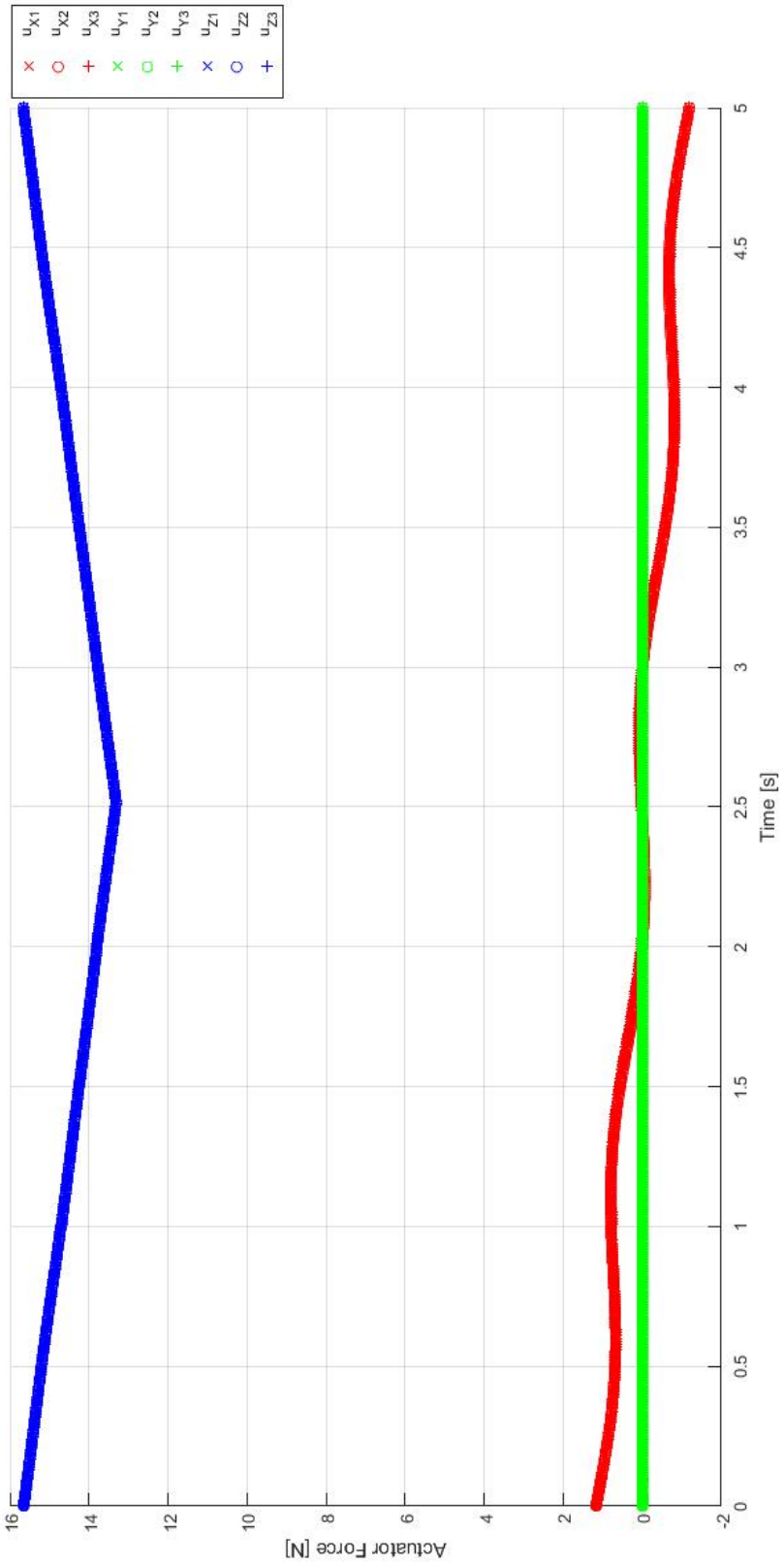
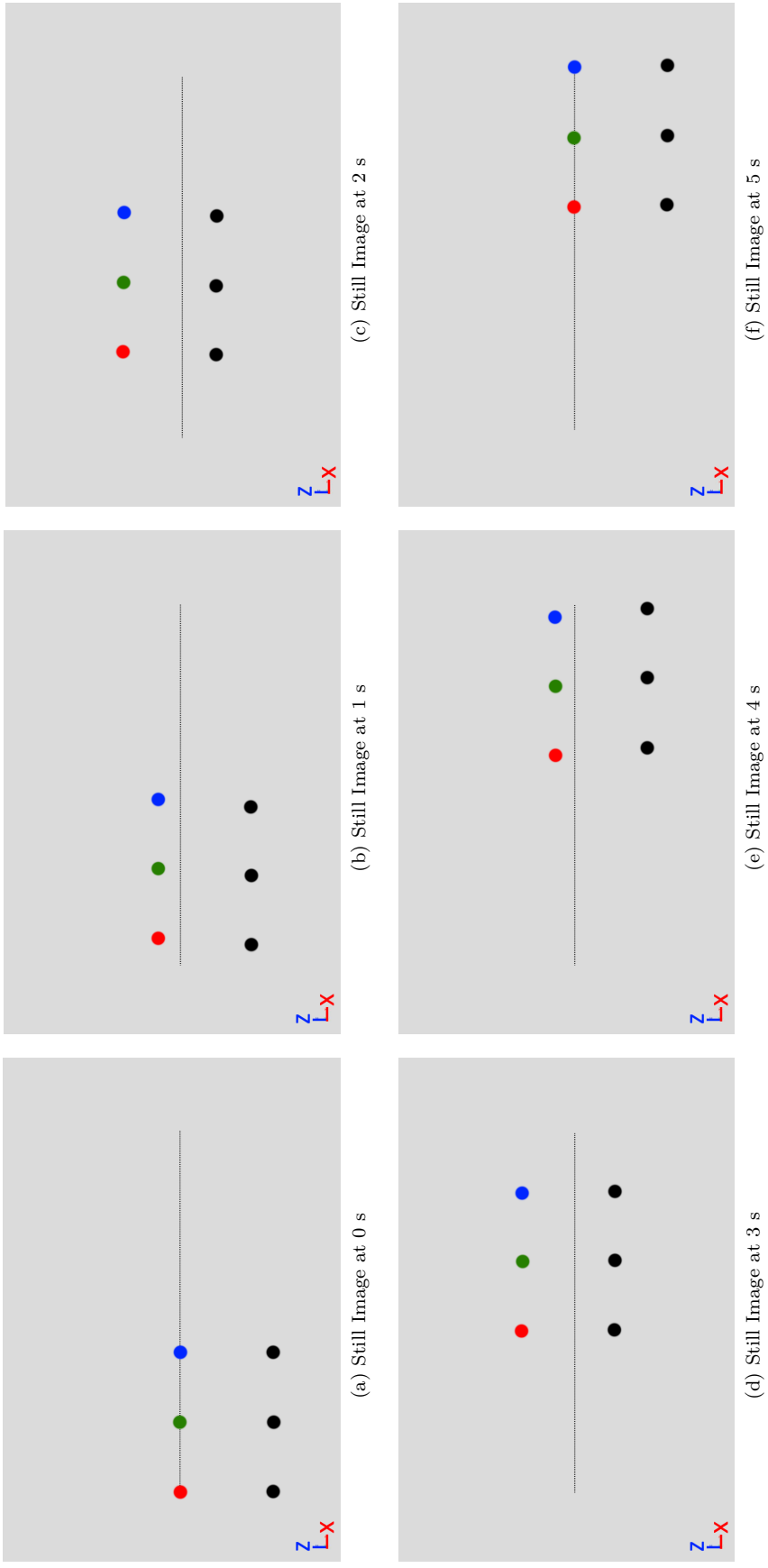


Figure 5.35: Evolution of System Inputs (Three Agents with Payload, Closed Loop with GDM Model)



(a) Still Image at 0 s (b) Still Image at 1 s (c) Still Image at 2 s

(d) Still Image at 3 s (e) Still Image at 4 s (f) Still Image at 5 s

Figure 5.36: Mission Visualization (Three Agents with Payload, Closed Loop with GDM Model)

Chapter 6

Conclusions

We conclude with a summary of the project's outcomes, evaluate performance and compliance with project objectives, and identify future work to advance the project.

6.1 Summary

Our project has focused on the application of MPC to the problem of cooperative payload transportation. Our primary goal has been to develop a novel MPC-based control architecture for an aerial multi-agent CPT scheme with a cable-suspended payload.

Problem Formulation: After surveying the literature on CPT schemes, summarizing the state of the art and explaining the key features of contemporary CPT solutions, we considered numerous CPT solutions in order to identify the necessary requirements for an aerial multi-agent CPT scheme with a cable-suspended payload. We applied these requirements to inform the design of a novel centralized triple-agent CPT scheme based on MPC theory. We analytically derived a mathematical model of the triple-agent system that represented the agents and segments of the payload as point masses governed by double-integrator dynamics. Using this model, we formulated a DHOCP in which the cost function was the sum of the squared agent thrusts. After identifying several physical constraints (such as payload rigidity and cable connectivity) and performance requirements (such as inter-agent collision avoidance and spatial obstacle avoidance) we formulated these as path constraints for the DHOCP.

Implementation Stages: We conducted a staged implementation of the proposed control architecture using the ICLOCS2 trajectory optimization package for simulations in MATLAB. We first considered the simpler problem of relocating a single agent while avoiding a spatial obstacle and then proceeded to formulate the relocation problem for a single agent with a cable-suspended payload, the dual-agent CPT problem, and the CPT problem for three agents with a cable-suspended payload.

Simulations: For each stage of the implementation process, we conducted at least two types of simulations to observe system behavior. We ran open-loop simulations, in which ICLOCS2 solved the DHOCP once and computed the evolution of the system's states and inputs. We verified the open-loop performance with simulations of the system in closed loop with ICLOCS2 by developing a GDM model of the system in Simulink, also allowing visualization of the system's physical behavior. To our knowledge, it was the first time that true MPC was implemented for aerial CPT schemes using cable suspension. In addition to the GDM model, we developed a closed-loop simulation using an ADM model. However, due to unforeseen numerical errors in Simulink only the single-agent simulation ran successfully. We also experienced a setback with the triple-agent system: ICLOCS2 could not converge to a feasible solution

for the original DHOCP with obstacle avoidance. To address this, we replaced three obstacle path constraints with a single obstacle path constraint, and softened this constraint by employing a slack variable weighted with an L_1 norm exact penalty function. Accordingly, we were able to demonstrate that the triple-agent system in closed-loop could indeed converge to a feasible solution and thus avoid a spatial obstacle, even though the computational runtime was impractically long.

Acquired Skills: The project has drawn on elements of optimization and numerical methods, mechanics, and optimal control theory. It may serve as a starting point for further work in CPT and robotic control because it has assisted in refining skills in model construction and simulation (particularly in MATLAB and Simscape). It has also helped to develop a working familiarity with ICLOCS2, now applicable to other optimal control problems.

6.2 Evaluation

Objectives: The project outcomes have met the objectives outlined in Section 1.3 of this report. We have implemented an optimal controller for a multi-agent CPT scheme that uses ICLOCS2 for trajectory optimization, simulating its performance in open-loop and closed-loop using Simulink. Through our work with the Simscape models we have also created visualizations of the CPT scheme in operation and used the results to verify the soundness of the open-loop simulation. We have achieved compliance with secondary mission objectives such as collision avoidance, eliminated the need for controller fine-tuning, and have been able to verify the optimality of controller inputs through the use of ICLOCS2.

Challenges: Despite achieving both objectives, a number of issues have affected the project's implementation.

While the mathematical models for the multi-agent CPT schemes appear reasonable, the performance of the implemented CPT schemes has needed improvement. In the tests that we conducted we focused on situations involving motion in a maximum of two dimensions and the presence of a spatial obstacle. However, it is worth noting that the triple-agent system's simulations were noticeably slow when compared with those of the single agent with a cable-suspended payload. It is possible that the DHOCPs for the three payload transportation systems considered in the project were posed in a format that the solver could not solve efficiently because of the presence of differential algebraic equations in the systems' models.

Another issue, pertaining in particular to the closed-loop simulations with an ADM model, was the presence of positive feedback as discovered in Section 5.2.2. Having designed the closed-loop simulations with the analytically derived and the GDM models from the same template in Simulink, we suspect that an improper configuration of the ADM model's sample times and the choice of a variable-step numerical integrator in Simulink may have destabilized the model's behavior.

Underlying the performance issues associated with both the open-loop and the closed-loop simulations we hypothesize that three factors could, individually or collectively, have adversely affected the soundness of the simulated models:

1. We have simplified the GDM model by approximating the agents and payload as point masses and the cables as distance constraints;
2. We have not considered the rotational dynamics of the agents and payload, such that we have assumed all roll, pitch, and yaw angles and their derivatives to be zero;
3. The models did not explicitly calculate the cable tension as a function of the separation between agent and payload, but rather the cable tension was implemented in ICLOCS2 as an input constrained within a narrow range of the hovering thrust of the payload and excluded from the DHOCP's objective function.

The first two factors are the most important sources of model inaccuracies. The simplifications to the GDM model have transformed each model from a coupled system of compound pendula (with each agent rotating about its cable attachment point, and the payload rotating about its cable attachment points), into a coupled system of simple pendula (with the payload split into multiple point masses, each rotating about their respective agents without a change in orientation). The latter configuration neglects the true moments of inertia for each agent and the payload, leading to a significantly reduced model of the torques experienced by each agent and the payload. Furthermore, omitting rotational variables from the models prevents analysis of rotational dynamics and the interactions between angular and linear kinetic energies. Consequently the models assume, unrealistically, that the agents and payload maintain a constant orientation with respect to the coordinates system.

The third factor could explain the presence of high-frequency oscillations in the simulations of cable tension, as seen in Figures 5.16, 5.17, 5.26, and 5.34. The DHOCPs have not included stiffness constraints on the rate of change in cable tension so the solver has converged to solutions in which cable tension oscillates very rapidly.

Note that we have assumed that the agents and payload do not experience headwinds or travel fast enough to experience significant drag forces. In real-life scenarios, the aerodynamics of a multi-agent system carrying a payload must be included in the system model.

Project Context: Despite these limitations, the achievements of the project should be considered in the context of the wider field. The literature relating to aerial CPT indicates that no one has yet considered how to use MPC techniques for trajectory optimization and compliance with performance objectives. There appears to be a reliance on mature but less sophisticated control techniques such as the PID controller. We note however that MPC techniques have been used with success in terrestrial CPT rigid grasp systems [24][25][35][36]. By means of this project, we have demonstrated the viability of applying MPC to a multi-agent aerial CPT system, thereby presenting an opportunity for the development of CPT schemes that are more energy efficient, more reliable, and safer.

6.3 Future Work

A number of extensions to the project may be immediately pursued, with the performance issues described in Section 6.2 worth addressing first.

Problem Formulation: A more accurate model of the three payload transportation system models should incorporate insights from the body of aeronautical engineering literature on the dynamics of sling-load systems (such as in [40] and [41]), representing the system as one or more non-point objects connected via moderately elastic cables to a single payload. While this would enhance the realism of the model by including rotational dynamics, it could lead to a more computationally complex problem. For example, the triple-agent system in (3.29) has 36 states, while a more accurate triple-agent system would have 48 states corresponding to three agents and one payload, each with twelve states $(x, \dot{x}, y, \dot{y}, z, \dot{z}, \phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi})$. Nevertheless, the need for path constraints for the payload's mass distribution would be eliminated by the latter model, so the resulting DHOCP could be solved more easily by ICLOCS2. We should then ask whether more than three agents could be viably included in an aerial cable suspension CPT scheme. If the relevant DHOCPs are computationally intractable for the optimization solver, a mechanically simpler rigid-grasp CPT scheme might be the only viable solution given the state of current optimization technology.

Model Improvements: For cases involving a payload, the effect of positive feedback in the closed-loop simulations with an ADM model should be eliminated. This issue could be corrected by using a fixed-step solver with smaller step sizes and reoptimizing the control inputs with greater frequency. To impose the condition of cable stiffness on the solutions found by the optimization solver, path constraints on the rate of change in cable tension should be implemented in ICLOCS2 (perhaps using finite difference methods). The aerodynamics of the system should also be explored, particularly situations where there are significant headwinds or when the payload experiences significant drag.

Operational Enhancements: Other operational assumptions about CPT schemes should be reviewed after the aforementioned issues are resolved. Implementing 'warm-starting' in ICLOCS2 should reduce the computational runtime of the closed-loop simulations by initializing the solver with the previous solution. Three-dimensional payload relocation with one agent and multiple agents should be implemented by modifying the intial and final state settings in the problem file for ICLOCS2. Active damping maneuvers by agents to prevent excessive payload swings should be explored. After confirming the safety of the models' behavior in simulation, the viability of the existing dual- and triple-agent schemes should be tested using suitable hardware. This would require the development of an interface between ICLOCS2 and the relevant software infrastructure (such as the Robot Operating System).

Control Paradigms: In the longer term, several possibilities for future research remain open. In our project we have considered a centralized CPT paradigm in which computation of control inputs occurs in one place, with full knowledge of all states. The disadvantages of the centralized paradigm are the reliance on communication channels and the greater computational complexity of the resulting DHOCPs. Paradigms in which the computation of control inputs is delegated to individual agents would be more appropriate for a real-world scenario, relying on either full communication (distributed control) or no communication between agents (decentralized control).

Beyond the issue of control architecture, a number of additional topics in CPT merit further investigation.

Formation Composition: In the current project we have assumed that the payload is manipulated by a formation of identical agents, however a richer class of CPT schemes would handle non-identical agents of different sizes, thrust capabilities, and hardware configurations. In hardware tests these considerations would be fundamental to the notion of system robustness: the possibilities of a hardware fault, mismatch between the agents' dynamical properties, or a breakdown in communications would immediately create a system of heterogeneous agents. There may also exist situations in which a heterogeneous agent formation is beneficial, for example when one or more humans are included in the formation. Such interactions would involve varying degrees of cooperation and command hierarchies among agents' controllers, so an analysis of multi-agent decision-making in the system would depend on concepts from game theory.

Non-Ideal Mechanical Behavior: The limitations of using an ideal model of the system might become apparent in hardware testing, so the model's accuracy could be enhanced by considering non-ideal mechanical behaviors, such as cable elasticity or payloads with non-rigid compositions (e.g. liquids and gels). Accounting for these phenomena would inevitably increase the complexity of the system model, so the balance between model accuracy and computational tractability would be a fruitful topic to investigate.

Each of these topics constitute a rich area of contemporary research in robotic manipulation and transportation, opening up avenues for further exploration of the CPT problem.

Bibliography

[1] S. Gupte, P.I.T. Mohandas, and J. M. Conrad, "A survey of quadrotor Unmanned Aerial Vehicles," in *2012 Proc. IEEE Southeastcon*, Orlando, FL, pp. 1-6. doi: 10.1109/SECon.2012.6196930

[2] F. Ruggiero, V. Lippiello, and A. Ollero, "Introduction to the Special Issue on Aerial Manipulation," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 2734-2737, Jul. 2018. doi: 10.1109/LRA.2018.2830750

[3] D. Shimer. (2018). *With Drone Deliveries on the Horizon, Europe Moves to Set Ground Rules* [Online]. Available: https://www.nytimes.com/2018/12/31/business/europe-commercial-drone-delivery.html/. [Accessed: Jan. 17, 2019].

[4] C. Dillow. (2016). *The Construction Industry is in Love with Drones* [Online]. Available: http://-fortune.com/2016/09/13/commercial-drone-construction-industry/. [Accessed: Jan. 17, 2019].

[5] M. Mazur. (2016). *Six Ways Drones are Revolutionizing Agriculture* [Online]. Available: https://www.technologyreview.com/s/601935/six-ways-drones-are-revolutionizing-agriculture/. [Accessed: Jan. 17, 2019].

[6] M. Blau. (2017). *Six Ways Drones Could Change Health Care* [Online]. Available: https://www.scientificamerican.com/article/6-ways-drones-could-change-health-care/. [Accessed: Jan. 17, 2019].

[7] B. Knight. (2017). *A guide to military drones* [Online]. Available: https://www.dw.com/en/a-guide-to-military-drones/a-39441185/. [Accessed: Jan. 17, 2019].

[8] F. Ruggiero, V. Lippiello and A. Ollero, "Aerial Manipulation: A Literature Review," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1957-1964, Jul. 2018. doi: 10.1109/LRA.2018.2808541

[9] Y. Nie, O. Faqir, and E.C. Kerrigan*, ICLOCS2.* London, UK: Imperial College London, 2018.

[10] The MathWorks Inc., *MATLAB R2017a.* Natick, MA: The MathWorks Inc., 2017.

[11] The MathWorks Inc., *Simulink 8.9.* Natick, MA: The MathWorks Inc., 2017.

[12] The MathWorks Inc., *Simscape 4.2 and Simscape Multibody 5.0.* Natick, MA: The MathWorks Inc., 2017.

[13] H.B. Khamseh, F. Janabi-Sharifi, and A. Abdessameud, "Aerial manipulation — A literature survey," *Robot. Auton. Syst.*, vol. 107, pp. 221-235, Sep. 2018. doi: 10.1016/j.robot.2018.06.012

[14] C. Gabellieri, M. Tognon, L. Pallottino and A. Franchi, "A Study on Force-Based Collaboration in Flying Swarms," in *Proc. 11th Int. Conf. ANTS 2018*, Rome, Italy, pp. 3-15. doi: 10.1007/978-3-030-00533-7_1

[15] Y.H. Tan, S. Lai, K. Wang, and B.M. Chen, "Cooperative control of multiple unmanned aerial systems for heavy duty carrying", *Annual Reviews in Control*, vol. 46, pp. 44-57, 2018. doi: 10.1016/j.arcontrol.2018.07.001

[16] M. Farina and G. Ferrari Trecate. Class Lecture, Topic: "Decentralized and distributed control: Introduction", EECI-HYCON2 Graduate School on Control 2012, Supélec, France, 2012.

[17] R. d'Andrea, "Guest Editorial: Can Drones Deliver?," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 3, pp. 647 - 648, Jul. 2014. doi: 10.1109/TASE.2014.2326952

[18] M. Gassner, T. Cieslewski, and D. Scaramuzza, "Dynamic collaboration without communication: Vision-based cable-suspended load transport with two quadrotors," in *2017 IEEE Int. Conf. Robotics and Automation*, Singapore, pp. 5196-5202. doi: 10.1109/ICRA.2017.7989609

[19] T. Tomic et al., "Toward a Fully Autonomous UAV: Research Platform for Indoor and Outdoor Urban Search and Rescue," *IEEE Robot. Autom. Mag.*, vol. 19, no. 3, pp. 46 - 56, Sep. 2012. doi: 10.1109/MRA.2012.2206473

[20] S. Tang and V. Kumar, "Mixed Integer Quadratic Program trajectory generation for a quadrotor with a cable - suspended payload," in *2015 IEEE Int. Conf. Robotics and Automation*, Seattle, WA, pp. 2216-2222. doi: 10.1109/ICRA.2015.7139492

[21] J. Gimenez, D.C. Gandolfo, L.R. Salinas, C. Rosales, and R. Carelli, "Multi-objective control for cooperative payload transport with rotorcraft UAVs," *ISA Trans.*, vol. 80, pp. 491-502, Sep. 2018. doi: 10.1016/j.isatra.2018.05.022

[22] N. Toumi, Decentralized Control of Multi-Agent Aerial Transportation System, Thuwal, KSA: KAUST, 2017.

[23] M. Tognon, C. Gabellieri, L. Pallottino, and A. Franchi, "Aerial Co-Manipulation With Cables: The Role of Internal Force for Equilibria, Stability, and Passivity," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 2577 - 2583, Jul. 2018. doi: 10.1109/LRA.2018.2803811

[24] A. Nikou, C.K. Verginis, S. Heshmati-alamdari, and D.V. Dimarogonas, "A Nonlinear Model Predictive Control Scheme for Cooperative Manipulation with Singularity and Collision Avoidance", arXiv: 1705.01426 [cs.RO], Nov. 2017.

[25] C.K. Verginis, A. Nikou, and D.V. Dimarogonas, "Communication-based decentralized cooperative object transportation using nonlinear model predictive control", arXiv: 1803.07940 [cs.RO], Mar. 2018.

[26] N. Michael, J. Fink, and V. Kumar, "Cooperative manipulation and transportation with aerial robots", *Auton. Robots,* vol. 30., no. 1, pp. 73-86, Jan. 2011. doi: 10.1007/s10514-010-9205-0

[27] P.O. Pereira, P. Roque, and D.V. Dimarogonas, "Asymmetric Collaborative Bar Stabilization Tethered to Two Heterogeneous Aerial Vehicles," in *2018 IEEE Int. Conf. Robotics and Automation*, Brisbane, QLD, 2018, pp. 1-7. doi: 10.1109/ICRA.2018.8460529

[28] T. Lee, "Geometric control of multiple quadrotor UAVs transporting a cable-suspended rigid body," in *53rd IEEE Conf. Decision and Control*, Los Angeles, CA, 2014, pp. 6155-6160. doi: 10.1109/CDC.2014.7040353

[29] B. Shirani, M. Najafi, and I. Izadia, "Cooperative load transportation using multiple UAVs", *Aerospace Science and Technology,* vol. 84, pp. 158-169, Jan. 2019. doi: 10.1016/j.ast.2018.10.027

[30] A. Tagliabue, M. Kamel, S. Verling, R. Siegwart, and J. Nieto, "Collaborative transportation using MAVs via passive force control," in *2017 IEEE Int. Conf. Robotics and Automation*, Singapore, pp. 5766-5773. doi: 10.1109/ICRA.2017.7989678

[31] R. Cotsakis, D. St-Onge, and G. Beltrame, "Decentralized collaborative transport of fabrics using micro-UAVs", arXiv: 1810.00522 [cs.RO], Oct. 2018.

[32] 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 Trans. Control Syst. Technol.,* pp. 1-8, Dec. 2018. doi: 10.1109/TCST.2018.2876518

[33] Z. Wang, S. Singh, M. Pavone, and M. Schwager, "Cooperative Object Transport in 3D with Multiple Quadrotors Using No Peer Communication," in *2018 IEEE Int. Conf. Robotics and Automation*, Brisbane, QLD, 2018, pp. 1064-1071. doi: 10.1109/ICRA.2018.8460742

[34] H. Lee, H. Kim, W. Kim, and H. J. Kim, "An Integrated Framework for Cooperative Aerial Manipulators in Unknown Environments," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 2307-2314, Jul. 2018. doi: 10.1109/LRA.2018.2807486

[35] H. Ebel and P. Eberhard, "Distributed Decision Making and Control for Cooperative Transportation Using Mobile Robots", in *Proc. 9th Int. Conf. Sensing and Imaging,* Shanghai, China, 2018, pp. 89-101. doi: 10.1007/978-3-319-93818-9_9

[36] R. Babaie and A.F. Ehyaie, "Robust optimal motion planning approach to cooperative grasping and transporting using multiple UAVs based on SDRE", *Trans. Inst. Measurement and Control,* vol. 39, no. 9, pp. 1391-1408, Apr. 2017. doi: 10.1177/0142331216640600

[37] V.G. Rao and D.S. Bernstein, "Naive control of the double integrator," *IEEE Control Systems*, vol. 21, no. 5, pp. 86-97, October 2001. doi: 10.1109/37.954521

[38] The MathWorks, Inc., *Documentation: ode23t.* (2019). [Online]. Available: https://www.mathworks.com/help/matlab/ref/ode23t.html

[39] E.C. Kerrigan and J.M. Maciejowski, "Soft Constraints and Exact Penalty Functions in Model Predictive Control", in *Control 2000 Conference,* Cambridge, UK, 2000.

[40] L.S. Cicolani and G. Kanning, "Equations of motion of slung-load systems, including multilift systems," NASA Ames Research Center, Moffett Field, CA, Tech. Report. NASA-TP-3280, 1 Nov. 1992.

[41] A. Cicchino and I. Sharf, "Three Formulations of Sling Load Dynamics for UAV Motion Planning and Control" in *Proc. ASME 2018 IDETC/CIE,* Quebec City, Canada, 2018, pp. V006T09A001. doi: 10.1115/DETC2018-85174

[42] K.J. Åström and R.M. Murray, *Feedback Systems: An Introduction for Scientists and Engineers.* Princeton, NJ: Princeton University Press, 2008.

Appendix A

Single Agent with Pendulum Dynamics Linearization

Consider the system described in Section 4.2, comprising a single agent with a cable-suspended payload. The state transition equation is presented in (A.1):

$$F(\overline{X}(t), U(t)) = \frac{d}{dt}\begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \\ z \\ \dot{z} \\ x_c \\ \dot{x}_c \\ y_c \\ \dot{y}_c \\ z_c \\ \dot{z}_c \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \frac{1}{m_{ag}}(u_x - F\tan(\alpha)) \\ \dot{y} \\ \frac{1}{m_{ag}}(u_y - F\tan(\beta)) \\ \dot{z} \\ \frac{1}{m_{ag}}(u_z - F - m_{ag}g) \\ \dot{x}_c \\ \frac{1}{m_{pl}}(F\tan(\alpha)) \\ \dot{y}_c \\ \frac{1}{m_{pl}}(F\tan(\beta)) \\ \dot{z}_c \\ \frac{1}{m_{pl}}(F - m_{pl}g) \end{bmatrix} \tag{A.1}$$

Consider the continuous-time system described by (A.1) and define the following variables:

- the state equilibrium point $X_e = [2.1, 0, 0, 0, 2.2, 0, 2.1, 0, 0, 0, 1, 0]^T$,
- the inputs $U_e = [0, 0, (m_{ag} + m_{pl})g]^T$,
- the new state variable $z = \overline{X} - \overline{X}_e$,
- the new input variable $v = U - U_e$.

We may linearize the system about $(\overline{X}_e, U_e)$ [42]:

$$\frac{dz}{dt} = Az + Bv \tag{A.2}$$

where

$$A = \frac{\partial F}{\partial \overline{X}}|_{\overline{X}=X_e, U=U_e}$$
$$B = \frac{\partial F}{\partial U}|_{\overline{X}=X_e, U=U_e}$$

We can then use a MATLAB live script to evaluate the Jordan form of A:

$$J = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\sigma_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\sigma_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{A.3}$$

where $\sigma_1 = \dfrac{\sqrt{918870}\, i}{180}$.

We note that since the off-diagonal entries of J are non-zero, the Jordan blocks corresponding to eigenvalues of A with zero real part are non-scalar and therefore the open-loop system is unstable [42].

Appendix B

Source Code

This project has an online repository at https://github.com/williamsdaniel888/ocplot.

Contents of Repository

- `/SingleAgent` – scripts and functions for open-loop and closed-loop simulations relating to the single-agent relocation problem.
- `/SingleAgentPayload` – scripts and functions for open-loop and closed-loop simulations relating to the single-agent payload transportation problem.
- `/DualAgentCPT` – scripts and functions for open-loop and closed-loop simulations relating to the dual-agent CPT problem.
- `/TripleAgentCPT` – scripts and functions for open-loop and closed-loop simulations relating to the triple-agent CPT problem.

Installation

A guide to installing and setting up the ICLOCS2 trajectory optimization package for MATLAB may be found at http://www.ee.ic.ac.uk/ICLOCS/Examples.html/. We recommend the use of the Opti distribution's Ipopt MEX file with ICLOCS2.

After cloning the repository, please ensure that the files for ICLOCS2 and Ipopt are placed in the root directory.