2018-02

Control of an Aerial Manipulator with an Attached Two DoF Arm

Staples, Marshall James

UNIVERSITY OF CALGARY

Control of an Aerial Manipulator with an Attached Two DoF Arm

by

Marshall James Staples

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF SCIENCE

GRADUATE PROGRAM IN MECHANICAL ENGINEERING

CALGARY, ALBERTA

FEBRUARY, 2018

Abstract

An aerial manipulator is the combined system of a robot arm attached to an unmanned aerial vehicle (UAV). The aerial vehicle increases the size of the robot arm's workspace in three dimensions, allowing the robot arm to execute manipulation tasks where it could not before. However, it is challenging to effectively control the position of the end-effector of a robot arm attached to a UAV due to the kinematic and dynamic coupling of the two systems.

The focus of this work is to develop an effective control algorithm to enable a aerial manipulator comprising an 2-DOF robot arm and a UAV to follow complex trajectories in 3D space as well as maneuver the UAV in coordination with the arm to place the end-effector in position and orientation suitable for tasks not possible today. To accomplish this task, a mathematical model of the coupled kinematics and dynamics of the UAV and robot arm are formulated. In addition, such model is used for the control of the aerial manipulator. The chosen control solution of an adaptive sliding mode controller for the robot arm is shown to be effective in simulation tests. With the proposed control architecture, the robot arm is able to track any 3D curve projected into its planar workspace while the UAV performs the needed movements to map the robot arm 2D trajectories into the desired 3D curves.

Keywords: MM-UAV, aerial manipulation, adaptive sliding mode control, 2-DoF Revolute Arm

ACKNOWLEDGEMENTS

I would first like to thank Dr. Alex for all of his support and guidance. He has placed countless hours into guiding me and my research and has edited my work for countless hours. I would also like to thank all of my lab mates for their conversation and support: Krispin Davies, Graeme Wilson, Diego Latorre, Mahmoud Mustafa, Jie Wang, Ashraf Abdelrahman, Jaime Garcia and Shadi Moghaddasi. I would also like to thank all of my other friends and family who offered their encouragement, advice, and emotional support.

Table of Contents

List of Tables

List of Figures and Illustrations

List of Symbols, Abbreviations and Nomenclature

Acronym	**Definition**
CoG	Center of Gravity
CoM	Center of Mass
HM-UAM	Highly Maneuverable Unmanned Aerial Manipulator
HM-UAV	Highly Maneuverable Unmanned Aerial Vehicle
MM	Mobile Manipulator
USAR	Urban Search and Rescue
w.r.t	With Respect To
VPP	Variable pitch propeller
VTOL	Vertical Take Off and Landing

Roman Alphabet	**Definition**
a, b, c, d	Quaternion vector values
$\hat{a}$	Generalized acceleration vector
B^{-1}	Control gain matrix
B_i	Sub-calculation vector for Coriolis acceleration
$\hat{c}_i$	Coriolis acceleration of link i
C_i	Sub-calculation vector for Coriolis acceleration
d_i	Position of joint i w.r.t CoM $(i-1)$
e	Generalized coordinate tracking error
E	True modeling error
f	Inverse dynamics control vector
$\tilde{F}_{est}$	Uncertainty Estimation Vector
F_i	Force vector i
$\hat{F}$	Generalized force vector i
$\hat{I}_i$	Generalized inertia of link i
$\hat{I}_i^A$	Generalized articulated inertia of link i
p_i	Position vector of i
q	Joint space inputs
q^*	Modified control input
Q_i	Applied joint torque for joint i

List of Symbols, Abbreviations and Nomenclature (Continued)

R	Rotation matrix
s	The sliding surface
$\hat{S}_i$	Spatial joint axis vector i
$\hat{T}_i^{\lambda(i)}$	Generalized spatial transform matrix
$\hat{T}_i^{j*}$	Reverse generalized spatial transform matrix
u_i	Joint axis vector i
$u(t)$	Control input vector
v_i	Linear velocity of i
$\hat{v}_i$	Generalized velocity of i
X_i	Cartesian coordinates of i
$\hat{Z}_i^A$	Effective generalized force on link i
$\hat{Z}_i$	Spatial bias force acting on link i

Greek Alphabet	**Definition**
α	Tait-Bryan XYZ Roll angle
β	Tait-Bryan XYZ Pitch angle
Γ	Adaption rate
γ	Tait-Bryan XYZ Yaw angle
ζ	State of the HM-UAM
θ	Robot arm joint position vector
λ_i	Parent link of link i
Σ_i	Frame of Reference i
Φ_i	Angular orientation of i
ω_i	Angular velocity of i

Chapter 1 - Introduction

Unmanned aerial vehicles (UAVs) have garnered ample attention in recent years and, the industry as a whole is experiencing rapid growth [1], [2], [3]. Drones (a commonly used term for military and hobbyist UAVs) are widely available from professional UAV developers and service providers and are increasingly being used for numerous commercial applications [2], [3], [12]. Typical commercial use of UAVs includes remote inspection tasks [9], cinematography [11], [14], photography, and photogrammetry [4], [5], [8], [14]. UAVs are also used by the military for automated surveillance purposes, counter terrorism [12], and other, more general aerial military missions [6]. Currently, the range of possible applications for UAVs is constrained by the need to avoid contact with their surrounding environment in order to prevent the likelihood of a crash. The potential to broaden the typical application of UAVs beyond simple non-contact tasks could revolutionize the UAV industry [7]. Possible uses of UAVs that interact with the environment include inspection and maintenance of power lines, transporting and delivering packages, search and rescue operations, operation of remote equipment such as well pads in the oil and gas industry, and performing tasks at heights that typically require ancillary support equipment, such as high-rise building window washing. In order to advance the UAV industry to the point where physical interaction with the environment is commonplace, modifications need to be made to current UAV technology. These modifications could range from mounting additional sensors or tools, such as manipulators and graspers, to changing the core design of UAVs to better fit-for-purpose to perform specific tasks that require interaction with the environment. The sub-set of UAVs equipped with additional tools to interact with the surrounding environment are often called unmanned aerial systems (UAS). More specifically, in what follows an UAV equipped with one or more robot manipulators (i.e., robot arm and

gripper/end-effector) will be referred to as an unmanned aerial manipulator (UAM). Aerial manipulation with VTOL (Vertical Takeoff and Landing) UAVs has become a field of interest in recent years e.g. [25], [29], [30], [31], [33], [36], [42]. Continuing research into the extension of the possible applications of UAMs is compelling, but there are a large set of challenges yet to overcome.

1.1 Motivation

Attaching robotic arms to controllable moving platforms is not a new concept. The area of research that deals with robotic arms attached to moving vehicles is typically known as mobile manipulation. Typical mobile manipulator (MM) platforms used include ground vehicles, ships at sea [15], and even space craft [53]. The research in MMs with such specific mobile platforms is extensive. However, using UAMs as MMs is a relatively newer area of research that has not been fully explored. The unique workspace of aerial vehicles, specifically rotor craft, creates new opportunities for mobile manipulation. The main advantages of aerial manipulation include the ability to work at height and the ability to traverse terrain that is difficult or impassible for ground vehicles.

Eliminating the need for equipment such as lifts and ladders, UAMs could be used for tasks as simple as window cleaning, changing light bulbs in vaulted ceilings, and hanging banners. Complex tasks can also be completed using UAMs. Specialized graspers and tools can be outfitted on UAS to complete more complicated tasks such as those found on a construction site. Welding, grinding, riveting, drilling, and assembling can all be accomplished using single or multiple UAMs outfitted with the proper tools. Operating either autonomously or under human control, UAMs can be used in dangerous environments removing humans from risk such as at height on a skyscraper or a pipe-rack in a chemical or nuclear processing plant.

Unlike unmanned ground vehicles (UGVs), aerial vehicles do not have to drive slowly over rough terrain and can bypass the difficult terrain by flying over it thus saving time and operation cost. The workspace of UAMs makes them well suited for tasks in remote locations, such as soil and water sampling. Soil and water sampling is a routine part of operations in many industries including oil and gas, mining, and forestry. UAMs could be used to perform checks and operate site equipment (e.g. turn valves), as would a human operator. However, instead of navigating winding roads and potentially spending hours commuting to specific sites, an operator could deploy the UAMs to quickly cross impassible or difficult terrain like rivers, swamp, and valleys. Similarly, UAMs can also be used to inspect and maintain remote infrastructure such as weather stations [18].

After a natural (e.g. earthquake) or accidental (e.g., explosion) disaster, quick response times are critical [17]. UAMs could play an important part of first response to disaster scenarios. With the ability to quickly fly over broken or unstable ground, UAV can aid first responders to quickly identify trapped victims before even reaching the disaster site. There are many advantages of using UAVs and UAMs over UGVs to assist on-the-ground first responders. The extra range of vision that UAVs have with their flight height advantage can help find survivors faster. Other advantages include being able to bypass unstable ground which could potentially cause further collapse or require slow and careful navigation. Situations where searching a building or a hazardous area may be dangerous for humans, such as unstable floors or structures, radioactivity, the presence of poisonous chemicals, or dangerous biological agents, can be safely addressed with the application of UAMs. Unlike a regular UAV, UAMs equipped with a robotic arm would have the ability to open doors [16], push buttons, activate switches, and move debris, allowing them to explore and interact with a wide variety of environments. In disaster situations,

UAMs can also serve people in distress by delivering communications, food, water, and medical supplies to trapped victims while they wait to be rescued.

Aerial manipulation is a complex and difficult task even under simple circumstances, and it is a much greater challenge to perform the many manipulation tasks that require a high degree of accuracy. One of the main hurdles is the combined and synchronized control of the aerial manipulation system as a whole. The coupled kinematics and dynamics between the robot arm and the UAV platform poses a significant control challenge. When the UAV's position changes, it creates reactive torques and forces at each of the joints in the robot arm. Furthermore, when the UAV moves (e.g., vibrates), the relative position, velocity and acceleration of the arm might change with respect to its goal trajectory because of the change in position of the arm's base (i.e, the UAV). Similarly, during arm operation, the arm produces reaction forces and torques of its own on the UAV, making it more difficult for the UAV to maintain its trajectory or pose (position and orientation) when flying (e.g., hovering). The reaction forces and torques that the equipped robotic arm and aerial vehicle apply to each other is part of the coupled dynamics between the two systems. Because of the coupled dynamics, active (e.g., cooperative) control is required for both systems during manipulation.

Another challenge in aerial manipulation is the coupled ground dynamics. While grasping objects, and until the object is no longer in contact with the ground the UAM is in some way coupled with the ground. The ground (surface) normal force on the object and the corresponding friction between the object and the surface are both external disturbances that are introduced to the manipulation while there is the ground coupling. One final consideration is the payload limitation of UAMs. Both the aerial vehicle and the attached manipulator will have payload

limitations as well as limitations on the arm's operational workspace and UAV's flight envelope based on the grasped object's mass and inertia.

1.2 Focus

This dissertation is focused on the control challenge of 'end-effector trajectory following' for aerial grasping and grasping and manipulation. The control of the manipulator following a given desired trajectory is a key component of aerial manipulation. The control of the aerial manipulation platform is also very important as this also influences manipulator control. For the purposes of this thesis, it is assumed that the trajectory that the UAM is required to follow has already been generated (provided to the UAM) and the remaining challenge is to follow that trajectory via the UAM's end-effector as closely as possible.

1.3 Scope

The general goal of this dissertation is to develop and test a suitable yet effective control system for a novel highly maneuverable Bi-tiltrotor with a pitch trim tail rotor UAV equipped with a two degree of freedom manipulator. To reduce the potential disturbances that the arm might exert on the UAV, the arm is constrained to only move within the vertical longitudinal plane of the UAV. This section outlines what is covered in this dissertation towards achieving such goal. A mathematical model of both the coupled kinematics and coupled dynamics for the UAM of interested is presented. In Chapter 4, the frames of reference for the system are introduced which includes the global frame, the aircraft's body frame (frame of the UAV), and the end-effector frame. The forwards kinematics of the combined UAM are derived. In Chapter 5, the corresponding mathematical formulations of the UAM dynamics is presented. The forces and torques acting on each link in the multi-body system are found using recursive Newton-Euler multi-body dynamics while the control strategy of the UAM is considered in Chapter 6. The

implementation of the control law as well as the Lyapunov stability proof is provided for the proposed adaptive sliding mode controller. The inverse kinematics for the UAM are also provided for the generic case when following 3D trajectories (which extend outside the vertical longitudinal plane of the UAV) with the end effector. This dissertation also involves the modeling of the UAV and arm in a simulation environment where testing of the control strategy can be preformed. The model is built in V-Rep, a commercial software coupled with the physics engine Vortex. The control strategy is written in a Matlab script and is used to control the UAM using a remote API built into V-rep. The performance of the control system for the agile Bi-tiltrotor manipulation system is tested in simulation and the results are shown in Chapter 8.

Chapter 2 – Literature Review

There has been a recent push in research to further develop the capabilities of UAMs. Examples of useful tasks that a UAMs can perform, to certain degree, include: perching a UAV to save energy and perform surveillance, in-flight refueling to extend its operation range, applying force to open a door or drill a hole, package pick up and delivery, and object manipulation. For these and other tasks, standard UAVs are not equipped with the tools to succeed. Thus, the typical aerial platforms (base UAV) such as helicopters and quadrotors are usually modified and retrofitted with additional devices to better perform the required tasks. Aerial manipulation is one of the most challenging yet desired capabilities of UAVs. Aerial manipulation tasks can be divided into several categories: slung load transportation, aerial grasping, and manipulation tasks using robotic arms.

2.1 Load Transportation

One of the first areas of interest in aerial manipulation includes the transportation of objects both as fixed and slung loads. Transporting an unknown object with an aircraft is a challenging task due to the fact the object being transported changes the dynamics of the UAS and affects the UAS in diverse ways. Both the mass and the inertial tensor of the UAV are changed in addition to an increase in necessary control effort to achieve sufficient lift/aircraft-maneuverability. When slung loads are being transported, it is possible that the length of the suspension is either unknown or the information available is wrong. In [19], an estimator was designed to estimate the length of the suspension of the load as well as the system states of both the slung load and UAV. The estimation of the length of the cable is performed using vision which assists in adapting the model of the slung load. It is also used to acquire state information of both the UAV and slung load. Slung loads are highly likely to generate oscillations during

normal aircraft maneuvers. A feed forward control system is used in [20] to avoid producing residual oscillations in the slung load while maneuvering. When a load is transported by cables with a single aircraft, there is limited controllability of the payload [20], and the positioning and orientation of the load are not directly controllable during transportation. In cooperative UAV transportation, there is the capability for higher controllability of the payload as well as greater lift capacity. In [20], static equilibrium of slung loads is achieved using swarm UAV configurations of varying UAV positions with respect to the slung load. The limitation of the proposed methodology in [20] is the inability to damp out small oscillations due to the small perturbations of the UAVs' positions while lifting the load.

Oscillations and swinging of the object during slung load transportation can occur, and the swinging of the load creates a changing load on the UAV. Aircraft trajectories that minimize the amount of the object's swing during transportation were generated in [21]. Optimal trajectories using an open loop method for a quadrotor UAV with a suspended load were generated by minimizing a specific dynamic programing function where the quadrotor slung load system was able to effectively follow the desired trajectory. In most papers dealing with the transportation of slung loads, the system dynamics are derived from the point of view of the UAV. However, if tasks are required of the slung load, difficulties in the control can arise because the suspended load coordinates do not explicitly appear in the mathematical formulation [22]. In [22] the UAS is modeled using the load as the base frame so that the position and orientation of the load are explicitly shown in the mathematical formulation. The UAV used in [22], is used for slung load trajectory tracking where the load is suspended by a rod. Such system is controlled using a discrete-time linear quadratic regulator (DLQR) which is successfully used to track 3D-trajectories with the slung load in simulation. When swarms of UAVs are used to

transport slung loads, they are not rigidly attached to the load. The distance each UAV is from the given load is fixed by the length of the rod/wire (assuming the rod/wire does not deform and is always in tension), but the position and orientation (pose) of the UAV is able to change relative to the slung load. This is not the case, however, when the load is rigidly attached to the UAV. This case occurs when the UAS has a gripper which is rigidly attached to the aerial platform. A group of quadrotors with rigid grippers was used to cooperatively lift objects in [23] were the attitude control was done using proportional-derivative control. In such systems the UAVs and the object can be considered as a rigid artifact thus simplifying the interactions with the object at the expense of potentially increasing the effects between the UAVs. In [23] the quadrotors are used to transport objects in diverse arrangements including a straight line, 'T', 'L' and cross arrangements to lift objects cooperatively.

2.2 Aerial Grasping

For an object to be manipulated or transported by a UAM, it first must be grasped. For most load transport tasks currently performed by UAVs, the load is already attached or loaded by a human operator. The next step in the development of UAM is to get the UAM to grasp the object for transportation without outside (e.g. user) assistance. The task of grasping an object while the aircraft is in flight is difficult due to the potential unstable flight dynamics present in autonomous aerial vehicles, their limited accuracy in position control, and the challenge of coupling a UAM with an object partially constrained by the ground during grasping. In [24], the stability bounds of a helicopter under PID attitude control while grasping an object on the ground was developed and verified. In [24], the closed loop stability limits for both the mass of the grasped object and its positioning are determined. In such paper, an under-actuated gripper was used to ease the control problem while taking advantage of the gripper's compliance in grasping

unstructured objects. Aerial grasping does not only involve picking up objects; it can also include placing them. In [25], structures are built using quadrotors with attached grippers. The authors developed an autonomous swarm system that had quadrotors flying to pick up magnetic links and assembling them into cubic structures by simply placing the links close enough to each other to self-align the links via their magnetic forces. Although precise placement of the links was reduced, the corresponding control mechanisms involved developing suitable control methods for the pick up, transportation and placement of the magnetic links. The flight controllers were derived from the aircrafts' dynamics linearized around hover and used a PID controller for both the quadrotor's position and orientation. To facilitate this task, the quadrotor does not grasp the magnetic links while in flight, but instead cuts thrust to the propellers and lands on the desired structural member to be grasped. Once landed the aircraft grasps the member reducing the required mid-air maneuvers. The cubic structure incorporates snap fit joints further easing the aerial grasping procedure. Even though the approach has been proven to be effective, there is a concern that this type of joint and aerial grasping will not be strong enough for use with larger structures or manipulate non-engineered objects.

Grasping large objects using UAMs has also been shown to be a difficult task [26]. The size, mass and geometric features of some objects makes it difficult for traditional grippers to grasp, and the change in center of gravity that these objects exert on the aircraft makes the control challenging. A unique UAM configuration was proposed in [26] and [27] to solve the problem of grasping objects that are large relative to the size of the UAM. The proposed UAM in [27] is a reconfigurable multilink planar multirotor aircraft, in which each of the rotors comprising the aerial vehicle is attached to an individual link that can rotate relative to the rotor/link next to it. As a result, the UAS can reconfigure its shape to grasp the object by

wrapping its links around the chosen object while maintaining a configuration suitable for flight. By reversing the configuration process that took place for the grasping, the UAM can release the grasped objects. This system is controlled using a combination of LQI and PID control techniques. The multilink system is assumed to change its configuration at a slow time scale, so that the aircraft dynamics can be neglected and thus, the entire UAM is considered as a single rigid body.

Grasping for UAMs is not limited to the manipulation/transportation of objects, graspers have also been used to perch a UAV at a vantage point to conserve energy while completing its given mission (e.g., surveillance). In [28], a quadrotor was equipped with a gripper on top so that it can perch beneath bars. A visual feedback control system was used to recognize bar-like objects which orientated and positioned the gripper beneath the bar so that the bar can be grasped.

Although aerial grasping has been usually performed while the UAM is in a hover maneuver and trying to maintain its position, in [29], high speed flight grasping has been investigated/performed using a micro-UAV where the UAM is used to grasp a cylinder while in forwards flight. The system uses an avian inspired gripper attached to a 1-Degree of Freedom (DoF) robotic manipulator. A minimum snap trajectory is generated for the aggressive grasping maneuver, and a PD-controller with a feedforwards loop for the dynamics is used to control the system.

Most of research in aerial grasping involves an underactuated gripper that is either mounted directly to the aerial platform (without a multi-link arm) or is attached to a single degree of freedom manipulator. While grasping with a gripper attached to a UAV is a useful task

for picking up and delivering objects in semi-unstructured environments, manipulation of the objects is highly limited.

2.3 Aerial Manipulation

In an attempt to overcome the limitations described above, in [30], the authors split autonomous aerial manipulation into 3 main phases: i) an approach phase, ii) grasping phase, and iii) a manipulation control phase. Furthermore, three main approaches to the set up of a controller for a UAS performing aerial manipulation has been proposed [30]. The first approach is to decouple control between the aerial platform (i.e., UAV) and the manipulator. In this form of control, which is currently the most popular approach used by researchers during its grasping phase, the aerial platform is in hover and its controller compensates for the disturbances caused or coming from the manipulator. The manipulator's controller, on the other hand, compensates for the disturbances due to the rotorcraft moving about its desired hover position. The second approach is to kinematically couple the controllers, where each individual system (UAV and arm) comprising the UAM can monitor the position and velocity of the other. The third approach involves coupling the UAV and manipulator controllers at the dynamic level. In this approach, the whole system (UAV + manipulator) are modeled via a single coupled mathematical framework.

Within the above mentioned controller setups, many different control schemes have been proposed to perform aerial manipulation. These include: PID [30], [31], Impedance control [33], [34], [35], Adaptive Sliding Mode Control [36], Variable Parameter Integral Backstepping (VPIB) controller [18], [37], Model Reference Adaptive Control (MRAC) [38] and Visual Servoing Control techniques in [39], [40], [41], [32], and [42]. These different techniques have been applied to a variety of aerial manipulation tasks such as: applying forces to an object/wall,

grasping and manipulating objects, and manipulator trajectory following while the UAV is airborne.

Applying forces to objects/walls through aerial manipulation can be use in many potential applications including inspection tasks such as ultrasound [42] which require contact, drilling holes [44], and moving objects without lifting [45]. In [45], a Thrust-vectoring Tri-Tiltrotor UAV is used to exert forces on a large box and push it using a 1-DoF revolute manipulator which is also used to help exert moment control. The advantage of the rotor tilting system when compared to UAS with no tilting rotors is that the UAV can maintain its attitude while exerting 'large forward-thrusting forces'. To control this system, researchers have split the control framework into two parts. The 'Free-Flight controller' (FF controller) and a 'Force-Moment Exertion controller' (FME controller). The FF is the controller that the system uses in free flight (no manipulation) using a gain scheduling PID-dD controller. The FME controller on the other hand was used while the UAM is in hover and applying a horizontal force to an object. The FME controller uses a similar mechanism as the FF controller but employs adaptive gains based on the rotors' tilt angles. The manipulator controller is dynamically coupled with the FME controller during manipulation. In an attempt to cope with the limited forces that the manipulator can exert on objects the authors in [44] extended the work in [45] with the purpose of drilling a hole in a wall. For this a mechanical nose grinder was attached to the 1-DoF manipulator. The ability to decouple the pitch-attitude control from the forward thrust and hover control has been shown to be a major advantage of the Tri-Tiltrotor aerial platform.

Combined grasping and manipulating objects is one of the primary areas of research in aerial manipulation. One of the control challenges in aerial manipulation is dealing with the reaction forces that occur during arm movement and grasping objects contact. In [33],

researchers attempt to characterize the coupled interaction forces between the manipulator and the UAV through modeling the dynamics of the manipulator and UAV while developing a PID controller to compensate for the dynamics. In such research, a quadrotor aerial platform was equipped with three serial chain manipulators, controlled via force feedback during manipulation. However, the three manipulators were used cooperatively to only grasp and manipulate objects positioned below the UAM. A coupled kinematic and a decoupled dynamic models of the complete system were used in simulation to ascertain an estimation of the expected forces and torques that occurred during manipulation. The experimental results for trajectory following for the manipulator showed high oscillations the authors believe are the result of the limitations of the PID controller structure used to control the position of the quadrotor and the manipulator.

To further investigate the limitations of the diverse controllers for aerial manipulation recently in [46], a gantry system with attached manipulators was developed. The gantry system was equipped with force and torque sensors that measure the disturbances created by the motion of the arm. In turn, the gantry system was used to simulate a UAV model developed in OpenRAVE in an attempt to avoid the potential for crashing expensive equipment. Subsequently, in [33], impedance control was used to measure the reaction forces due to the arm's motion while peg-in-hole insertion tasks were attempted. Impedance control feedback was used to control the dynamic interaction between the manipulator, its environment, and the aerial platform. The gantry and manipulator system developed in [46] was used as the experimental platform. As with typical developments, during manipulation, the aircraft was restricted to be in hover, while the rotational and positional velocities and acceleration of the aerial vehicle were neglected in the dynamic model for the manipulator controller, as it was assumed that the aerial

platform would remain in hover. The forces and torques exerted by the manipulator on the rotorcraft were treated as external disturbances as the controllers were dynamically decoupled.

Despite advancements, UAMs have been restricted to using simple manipulators to limit the effects of the robot arm on the UAV operation. In an attempt to extend such capabilities, the authors in [34], investigated the interactions between a hovering helicopter and a multi-DOF attached manipulator, as well as the interaction between the UAM and fixed environmental objects. In such work, a 7-DoF revolute robotic arm was chosen as the manipulator and a vertical pole was used as the fixed environmental object to be grasped. As in other works, an impedance control strategy was selected as the manipulation controller to compensate for the interaction forces between the UAV and the fixed objects while using a pre-set UAV + object stiffness parameter. A transfer function for a simplified model of the UAV and Manipulator was created and used to create a controller for the helicopter using pole placement techniques. To ensure the stability of the controller, limitations on the maximum movement of the UAV's center of gravity (CoG) were used. Furthermore, to ensure that the arm does not exceed these limitations, the manipulator controller used a null-space control strategy to keep the arm's CoG on the center axis of the helicopter. To enable this the control scheme coupled the kinematics of both the arm and the UAV but decoupling the dynamics in the control. This system was also tested with a controller that was decoupled kinematically and dynamically and found that, if the controller is decoupled kinematically and dynamically, diverging phase circles occur.

In [38], the contact stability challenge and load estimation for a quadrotor with two 4-DoF arms was addressed. The system used a gain scheduling PID controller combined with an Model Reference Adaptive Controller (MRAC). The authors in [38] used the MRAC architecture to change the control loop gains to compensate for the change in the moment of inertia of the

UAM during the operation of the arm. The MRAC has been proven to be stable in simulation if and when Lyapunov stability theory is used.

A unified controller for UAMs is presented in [36]. Until this work, a unified controller had not been implemented for a UAM with a multi-DoF manipulator. An adaptive sliding mode controller was used to control a quadrotor equipped with a 2-DoF revolute-revolute robotic arm. The controller was chosen to deal with uncertainties (previously assumed negligible) such as modeling errors, and measurement noise and bias. The dynamic model of the UAM considered the coupled dynamics between the arm and the aircraft. The developed unified controller has been shown to be a successful approach to autonomously pick up objects, and transport them, followed by their placement on a given surface.

Although somewhat successful, the developments presented above only focus on grasping objects without arm path planning. A challenge in working toward autonomous manipulation is generating effective cooperative UAV and robot arm trajectories. In [32], [39], [40], and [42], Image Based Visual Servo (IVBS) is used to generate reference positions and velocity trajectories to grasp and manipulate objects.

In [32], a hexcopter is used as the aerial platform and a multi-DoF robot arm is used as the manipulator. The developed controller uses visual servoing combined with an adaptive sliding mode controller to enable the UAM to adjust for uncertainties. The controller considers both the dynamics and kinematics of the manipulator and aerial platform to be coupled. However, the Coriolis effects generated by the movement of the aerial platform (UAV) are omitted in the dynamic model.

In [40], an on-board eye-to-hand visual servoing system is used to generate velocity commands to both the UAV and the manipulator. A new visual error definition is used to

generate the system's task velocity commands. An Integral Backstepping low-level controller for the manipulator was developed to follow desired/computed velocity commands via the UAV and the manipulator coupled model.

More recently, enhanced UAM pick and place experiments were reported in [47] using a quadrotor. Using differential flatness in an attempt to improve the robot arm trajectories, the control of the system is decoupled into two independent controls: one controlling the position of the center of mass of each of the arm's links, and the second concerning the attitude of each link. Lyapunov convergence proofs were developed for each of the subsystems. This control strategy, although somewhat effective have only provided limited manipulator trajectory following results while improved results are provided for the position and orientation trajectory for the aerial platform. In most of the literature for aerial grasping, the focus of trajectory following has been on the positioning and orientation of the aerial platforms. There are very few work that address the positioning of the manipulator during its operation, so it is difficult to compare the effectiveness of the various controllers that have been proposed for the manipulators comprising the UAM.

Another of the control challenges of aerial manipulation is finding the bounds of stability for the given UAM for diverse aspects such as the movement of the overall UAM's CoG [30], and the limits of external forces that the UAM could potentially overcome [32]. Through knowing the bounds of stability, the limits of the controller can be understood. The stability of a Cartesian Impedance Controller for a quadrotor equipped with a 3-DoF arm subjected to external forces is analyzed in [32]. Such controller has been tested under two different cases - one where the external force is applied to the body of the UAV during manipulation and, in the second case, where the disturbance force is applied to the end-effector. The proposed controller has been

partially effective as the proposed controller is very specific to the particular UAM of interest with little generalization to other UAMs of interest with little generalization to other UAMs.

In [30], the stability bounds for the movement of the CoG of a helicopter UAV using a PID controller was investigated. The CoG and Inertial tensor of the helicopter was disturbed using a manipulator with a given added mass. It has been found that if the frequency of the interaction forces between the manipulator and the aerial platform are close to the frequency of the UAV movement, then unstable oscillations in the position of the aerial vehicle can occur.

As mentioned before, most aerial grasping, manipulation and trajectory following tasks have been performed while the UAV is in hover which limit the potential application of UAMs to the manipulation of static objects. In an attempt to handle dynamic objects, some researchers have considered the control of UAMs in motion. Using the motion of a UAV with a 3-DoF manipulator, the tracking of a helicoidal trajectory was attempted in [48]. Due to the associated complexities, the author in [48] constrained the UAV to a specific set of maneuvers. The UAV trajectory circles about a center axis while changing its elevation simultaneously the manipulator tries to keep the end effector on a given trajectory on the above-mentioned center axis. This trajectory tracking was performed using an adaptive sliding mode controller. Gaussian noise was added to the measurement signals from the states of the UAM. The dynamics controller model incorporates both the UAV and the manipulator and includes the coupled dynamics between the arm and aerial platform. While this work demonstrated that the control of both the manipulator and the UAV working in concert is achievable, most of the control effort was performed by the UAV and minimal trajectory corrections are performed by the manipulator. Despite these claims, the results and data reported for the trajectory following have been shown at a resolution that makes it difficult to analyze how well the trajectories are followed by the end-effector.

In an attempt to anticipate future interactions between the UAV and the robot arm comprising a UAM and cope with the problems experienced by researchers for effective control researchers have explored the use of Model Prediction Control (MPC). The first online and on board Nonlinear Model Predictive Controller (NMPC) of a UAM is presented in [31]. The aerial platform used is a multi rotor vehicle (octocopter) and is equipped with a 3-DoF serial manipulator. Simulations of waypoint following using both the manipulator and UAV was demonstrated to be successful. However, the results for the end effector following are presented on the scale of meters. This makes it difficult to discern how effective the UAM is at following the desired trajectory.

In an attempt to provide UAMs with diverse ways to perform tasks and solve some of the challenges generated due to the interaction between the UAV and the arm, researchers have explored the use of redundant robot arms. Performance optimization of a UAM involving a helicopter and a 7 DoF manipulator was investigated in [41]. Visual servoing combined with an impedance controller is used for both control and state estimation of the UAM. The performance of the optical localization is optimized by decreasing the time lag in sensor data by fusing the visual servo data with other state sensors. Various available and well-known compensations for the visual delays are applied to the system with relatively good positional error.

Throughout this section, it is repeatedly observed that few results are available that demonstrate the extent to which the end-effector is able to follow a given trajectory. This is due to the fact that the focus has been on grasping, and on the orientation and positioning of the aerial platform while the results for the positioning of the end-effector have been minimal. Most trajectory following tasks have been performed while the UAM is in hover, and very little work has been performed for grasping and manipulation tasks while the UAM is in motion.

2.4 Discussion

The field of aerial manipulation is presently in its early phases of development. Planar multi-rotor UAMs (i.e., multirotor vehicles such as quadcopters, and hexcopters with all rotors operating in the same plane), and helicopters have been predominantly the aerial platforms of choice. However, the array of manipulators used in aerial grasping and object manipulation has been diverse, and a wide range of robot arm aspects have been explored, including varying the number of DoF, the type of manipulator (e.g., serial link, parallelogram structure) and their design. The first efforts in aerial manipulation were focused on transportation of objects, and grabbing objects with simple grippers. The field of aerial manipulation then progressed to manipulation tasks using multi-link robotic arms where the tasks performed by the arms have been numerous (e.g., applying forces to surfaces and objects, grabbing and manipulating objects).

As mentioned in Section 1.2, the focus of this dissertation is on trajectory following of the end-effector comprising UAMs. Helicopters and quadrotors have been the primary aerial platforms used for trajectory following tasks. In grasping and trajectory following, most of the aircraft have been controlled using a PID feedback controller, and the manipulator has predominantly been controlled via visual servoing, PID controllers, impedance controllers, and sliding mode controllers.

There are many challenges left to be solved in the realm of aerial manipulation before the full potential of aerial manipulators can be realized. From the literature review, some of the identified challenges include:

1) Solve the closed chain mechanism problem when UAMs interact with objects that are partially constrained on the ground.

2) Develop new sensing tools to better perceive the complexities associated between the interaction of the UAV and the arm. This is especially important when developing and using UAMs in unstructured environments.
3) Develop/improve methodologies to enhance the ability to follow aerial trajectories via the combined motion of the UAV + Manipulator.
4) Develop a standard with which to rate how well an end-effector is following a trajectory in any given aerial manipulation.
5) Increase the speed in which manipulation tasks can be performed.
6) Develop controllers that can fully utilize both the movement of the UAV and the manipulator to successfully complete manipulation tasks.
7) Develop control systems capable of completing complex tasks that require precise wrenches applied to the end-effector such as opening doors and operating equipment, and
8) Solve the complexities involved with external disturbances applied to the UAM such as wind gusts and other aerodynamic effects such as the ground and wall effects that might be generated when the UAM is in close proximity to objects/terrain.

Chapter 3 – Problem Definition

The focus of this dissertation, as mentioned in Section 1.2, is on the control challenge of end-effector trajectory following during aerial manipulation while the UAV performs any needed aerial maneuver (e.g., pitch hover) to ease or make possible the grasping process (e.g., inserting the arm inside a hole to retrieve the object). From the wide array of challenges identified in Chapter 2, the following two challenges have been identified to be the next logical research steps to be addressed:

1) The ability to follow trajectories in 3D space using the combined motions of the UAV + Manipulator.
2) Developed needed controllers that are robust and can fully utilize both the movement of the UAV and the manipulator to precisely and completely achieve desired robot arm trajectories.

The above two challenges were selected as they comprise the most logical next step in the progression towards achieving full aerial manipulation. By increasing the accuracy with which an aerial manipulation system (i.e., UAV + robot arm) can follow trajectories with the end-effector, tasks that are currently impossible due to the requirement for higher precision manipulation such as turning a door knob to open a door in an Urban Search and Rescue (USAR) operation can be performed. Currently, most tasks completed in aerial manipulation do not require high or accurate precision. Examples of such tasks include applying forces to surfaces and pick and place tasks. The pick and place tasks, for example, have been partially solved by using compliant grippers to overcome the low accuracy of the of the end-effector's positioning during aerial object manipulation. Although the developed approaches have been shown to be

effective, such approaches cannot be applied if objects need to be handled with care (e.g., bomb defusal).

To date, there has been minimal work performed in aerial manipulation that uses the motion of both the UAV and manipulator to follow trajectories. Typically, when aerial manipulation is performed, it is done while the UAV is assumed to be executing a perfect (horizontal) hovering maneuver, near zero velocity.

Manipulation has been usually performed while the UAV is in hover since the UAV and the manipulator are kinematically and dynamically coupled and performing tasks while in hover reduces the needed control effort. Within the published literature, there are reported cases where aerial manipulation is performed while the UAV is not in hover; however, in such cases, the task/operation is limited/constrained, and the tasks often involve the UAV only moving along the vertical plane along the UAV's longitudinal axis. To the author's knowledge, aerial manipulation has not been performed while the aerial platform is yawing, or moving in the lateral direction which are maneuvers that are required in diverse operations including aerial manipulation in confined spaces (e.g., collapsed building during USAR operations). The challenge of controlling the end-effector to follow 3D trajectories using both the movement of the UAV and the manipulation still needs to be addressed.

Along with defining the chosen challenges to be solved, one must also select both an aerial platform and a manipulator type, as such systems will impose specific requirements, options and guide the development of the control effort. From Chapter 2, it was identified that most researchers use quadrotors or helicopters as the aircraft of choice. The reason for this is the availability of such platforms. The other less common aerial platforms used are tri-copters, hex-copters, and octocopters. Although somewhat effective and having greater payload carrying

capacity these aircrafts have limited maneuverability that prevents the better usage of any given robot arm mounted on the UAV. For this dissertation, a highly maneuverable thrust vectoring bi-tiltrotor with a pitch trim horizontal tail rotor (Figure 3.1) (also referred to as a HM-UAV and described in Chapter 4) was chosen as the aerial platform.

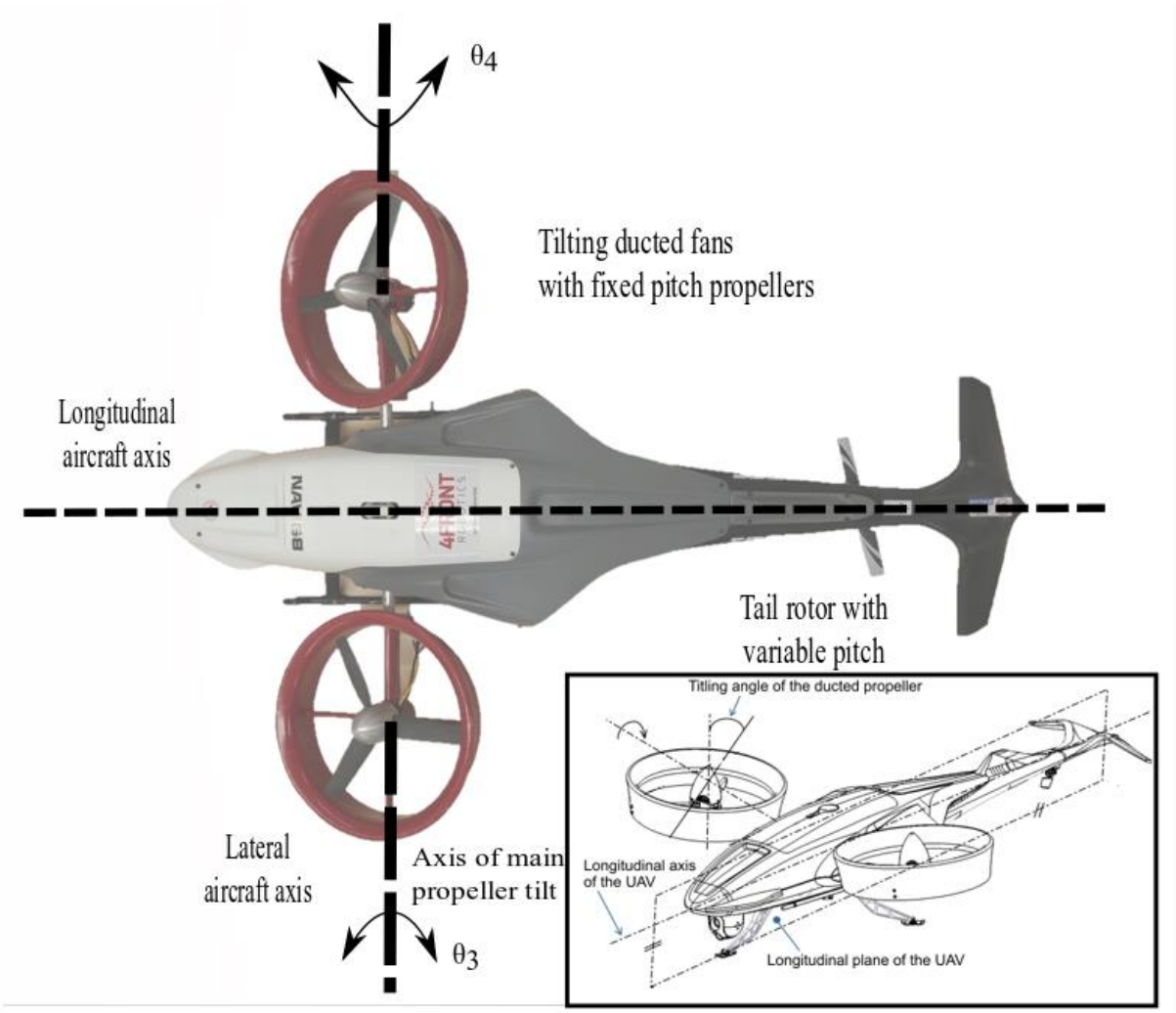


Figure 3.1 : HM-UAV an Agile Bi-Tiltrotor developed by 4Front Robotics Inc.

The reason for selecting such an aircraft is due to its capabilities to perform maneuvers that other aircraft cannot perform, such as controlled pitch hover, and forwards and backwards

movement without pitching. It is envisioned that those characteristics would enable and enhance better robot arm control for following 3D trajectories. To the authors' knowledge there have not been any successful aerial manipulation tasks or accurate trajectory following performed with any given HM-UAV using a robot arm with more than 1-DoF. This is due to the fact that such task has been a significant challenge to achieve. However, the goal for this thesis is to enable enhanced aerial manipulation. Therefore, the chosen robotic arm is a planar, 2-DoF revolute-revolute robotic arm. To ease the complexity of the problem at hand, the arm will be constrained to operate along the longitudinal plane of the UAV defined as the plane that cuts the UAV along its longitudinal axis into two halves as illustrated in Figure 3.1.

3.1 Problem Statement

As described above, the focus of this dissertation is to develop an effective control strategy to enable an aerial manipulator to follow trajectories in 3D space using the proposed UAV and manipulator. To accomplish this task, the combined motion of the UAV + manipulator will need to be controlled during the entire manipulation process. For example, for the arm to follow a curve in 3D, the UAM as a whole might need to change its attitude (position & orientation) to follow the trajectory. That is, because the chosen manipulator is constrained to operate in the longitudinal vertical plane of the UAV and cannot follow trajectories outside of this plane there is a need to enable a coordinated motion between the UAV and the arm. Therefore, to follow 3D trajectories, the manipulator must work in concert with the motion of the UAV. Furthermore, the yet to be developed controller will need to be able to cope with unknown loads on the end-effector to demonstrate that the UAM's control strategy can manipulate objects

of a priori unknown mass (a typical task in real applications). Thus, the specific problem statement is as follows:

3.2 Constraints

Design and implement a controller capable of tracking a reference trajectory in 3D space for a highly maneuverable thrust vectoring bi-copter equipped with a pitch trim horizontal tail rotor and a 2-DoF revolute-revolute planar arm under an unknown load attached to the end effector.

Due to the complexities associated with the research problem and in an attempt to ease the problem, a number of constraints will be imposed. The following is a list of the seven main constraints that help define the scope of the problem addressed in this dissertation:

1. Disturbances due to wind gust will not be considered. That is, the UAM will be considered to be always be operating under calm wind conditions.

2. The UAM will be considered to be operating far from any aerodynamic ground and wall effects, and disturbances due to ground and wall effects will be considered negligible.

3. Trajectories that the end-effector will be commanded to follow are in the immediate vicinity of the UAM, where path planning of the UAV is not necessarily required.

4. The generated trajectories that the UAM is commanded to follow will only consider the position of the end-effector and not its orientation. That is, in this work the grasping part is not considered.

5. The tasks that the robotic arm performs will not include its interaction with objects in the surrounding environment.

6. The unknown load that can be imposed at the end-effector (assuming a grasp has been completed) will be modeled as a point mass.

7. To enhance reaction time, the HM-UAV (Figure 3.1) will be assumed to use variable pitch propellers instead of fixed pitch. In this way the UAV is assumed to be capable to of changing the magnitude of the thrust from the propellers quickly.

3.3 Assumptions

To further reduce the high complexity of the problem at hand the following set of eight assumptions will be used throughout the research and development work.

1. The arm does not affect the downwards airflow around the UAV and it is assumed that there is no aerodynamic disturbance (forces) affecting the robotic arm. Furthermore, it is assumed that any downwash airflow generated by the UAV's propellers and hitting the arm does not produce any significant airflow ground and wall effects affecting the control of the UAV.
2. It is assumed that the magnitude of the thrust generated by the UAV's rotors can change relatively quickly when compared to the frequency of the controller. For this, it is furthermore assumed that the thrust generated by the propellers is controlled using variable pitch rotors, and having a low moment of inertia that their pitch angle can be changed quickly.
3. It is assumed that the propellers can be modeled in simulation as thrust vector acting along the center axis of the rotor pod. This assumption is made to reduce the complexity

in modeling. It is assumed because of the low change in angular inertia of the propellers during operation (because the thrust is controlled using variable pitch not speed), and the relatively small maximum angle of tilt for the rotors during operation that this assumption is valid.

4. The joint angles for the rotor joints are able to track the desired direction of thrust with minimal lag. This assumption is made because it is assumed that the change in position for the rotor joints will be relatively small.
5. It is assumed that any joint friction within the arm or UAV is negligible.
6. The links comprising the system and the UAV are assumed to be rigid and non-deformable.
7. The joint torques for the robot arm are controllable and the HM-UAM states (position, velocity, and acceleration) are observable.
8. The sensor data that the controller takes from the UAV's and robot arm's sensors is assumed to be accurate and error free.
9. The UAV and attached arm are assumed to be symmetric along the longitudinal vertical plane of the UAV.
10. The physical engine is perfectly representing the state of the vehicle when communicating with the controller.

3.4 Proposed Solution

In order to solve the proposed problem statement, based on the assumptions and constraints, a cooperative framework will be used to compel the end-effector to follow the given trajectory.

An adaptive sliding mode controller is proposed as the control system for the UAM. The controller will use a coupled dynamic model that will feed-forward the expected generalized forces (torques and forces) in order to make the nonlinear UAM system linear. The coupled kinematic model for the arm and the UAV will operate such that the combined system (UAV+ manipulator) will work together and cooperatively to follow the given trajectory.

The proposed solution will comprise a series of four generic steps in which each step will build upon the previous one.

STEP 1) First, a forward kinematic model of the system will be developed. This model will include all the frames of references for all the various links in the HM-UAM. An inverse kinematic model will also be developed so that the desired end effector and UAV positions can be translated into desired positioning for each of the joints connecting the links.

STEP 2) Develop a mathematical model of the dynamics of the HM-UAM system. This model will include the coupled dynamics between both the UAV and the manipulator. This dynamic model will be used in the controller development for both the UAV and the manipulator.

STEP 3) An adaptive sliding mode controller will be used to adapt to any modeling parameter errors. An adaptive sliding mode controller was selected because it does not rely on direct cancellation of the dynamics of the system, so the controller provides a robust response to unmodeled dynamics as well as unknown loads.

STEP 4) Develop a model of the system in a simulation environment. The purpose of this step is to create a system upon which testing can be performed and results analysed. The simulation environment will be developed in an existing simulation software, where information/data generated from the UAV and Robot arm CAD models can be used (e.g. location of the components CoM). Figure 3.2 illustrates these steps and showcases the flow of information between the models.

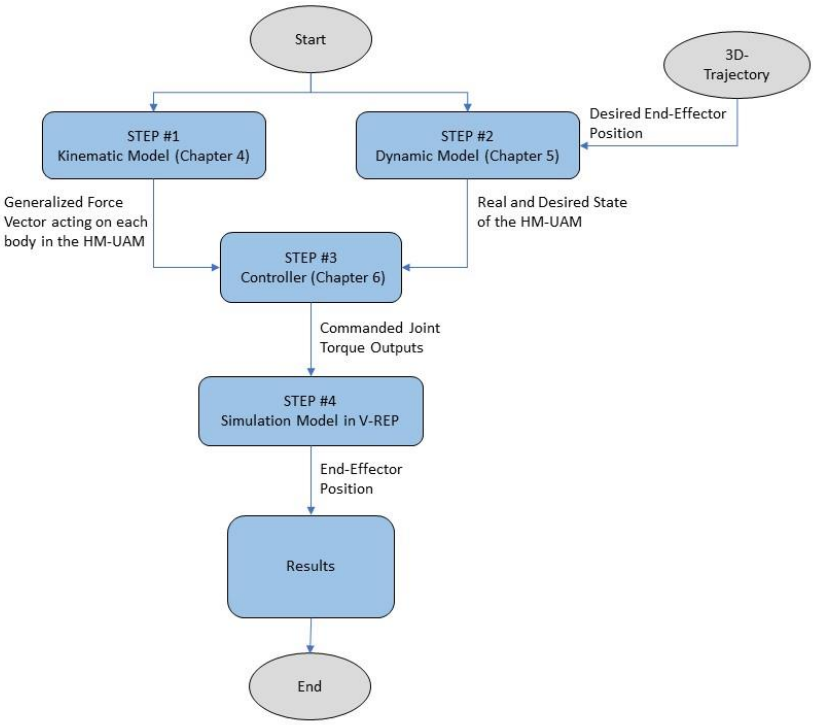


Figure 3.2: Design UAM Control Development Flow Chart.

3.5 Proposed Contributions

Following the steps outlined in Section 3.4, an effective control strategy/algorithm for a higly maneuverable UAV with an attached planar revolute-revolute manipulator under parameter and

dynamic uncertainties for following 3D positional trajectories will be achieved. The expected three main contributions from this work are listed as follows:

1. Develop an effective control strategy for a HM-UAV having an attached 2-DOF manipulator that accounts for both the coupled kinematics and dynamics between the UAV and arm using a properly formulated aerial manipulator mathematical model.
2. Develop a control strategy for the combined UAV + manipulator system suitable for following complex 3D trajectories not possible today. Hence complex 3D trajectories are defined as trajectories where the cooperation of the UAV and arm are required.
3. Develop a suitable performance analysis tool to assess the ability of the end-effector to precisely follow pre-defined trajectories in 3D space.

To the author's knowledge, the combined control of a HM-UAV with a multi-degree of freedom arm (a robotic arm with 2-DoF or more), has not yet been performed. Manipulation tasks have not been performed before in literature using highly maneuverable aerial vehicles. Currently, examples of HM-UAMs using Tri-Tiltrotors as the aerial platform have involved using single DOF arms to push on objects, and have not involved trajectory tracking or grasping. As seen in the literature review, the results for trajectory following have not been presented in a comprehensive manner. This dissertation presents a quantitative performance analysis for following 3D trajectories with a planar manipulator on a HM-UAV and records a comprehensive set of test results.

Chapter 4 – Description of the UAV.

This chapter describes the tiltrotor bi-copter which employs a horizontal tail propeller used in the context of this thesis. The tail propeller is mainly used to trim the aircraft's pitch attitude. It is not needed for lift but can serve such task in cooperation with the two main propellers, if and when needed.

In what follows, this UAV will be referred as to the Highly Maneuverable Unmanned Aerial Vehicle (HM-UAV). When equipped with an aerial manipulator it will be referred to as the Highly Maneuverable Unmanned Aerial Manipulator (HM-UAM).

4.1 HM-UAV

The HM-UAV was developed in Calgary by Dr. Alex Ramirez-Serrano (Patent #: PCT/CA2016/051197) for operations in confined spaces, particularly for Urban Search and Rescue (USAR) missions.

The HM-UAV is a highly maneuverable Vertical Take-Off and Landing (VTOL) UAV with a configuration that consists of twin-ducted fans flanking a central fuselage and one zero lift horizontal tail propeller mounted near the tail unit as shown in Figure 4.1 and Figure 4.2. To increase the reaction time and pitch control maneuverability of the HM-UAV, the tail propeller uses a variable pitch propeller (VPP) system. The main propellers can also employ a VPP but in this thesis we consider them to use fixed pitch propellers.

The HM-UAV uses two counter-rotating propellers that can tilt individually (fore and aft) enabling it to effectively perform the required maneuvers and cope with disturbances such as wind gusts. The optimized shroud design protects each propeller increases the propeller's thrust, and thus enables fifty to sixty percent more payload capacity when compared to similar size UAVs. These tilting ducted-fans are used to control the aircraft's lift, yaw, and roll in flight,

thereby eliminating conventional wings and ailerons. Although the HM-UAV (shown in Figures 4.1 and 4.2) uses a dihedral angle on the main tilting propellers to reduce any ground and wall airflow effects on the UAV, the UAV that is modeled in simulation does not have a dihedral angle (as ground and wall effects are assumed negligible).

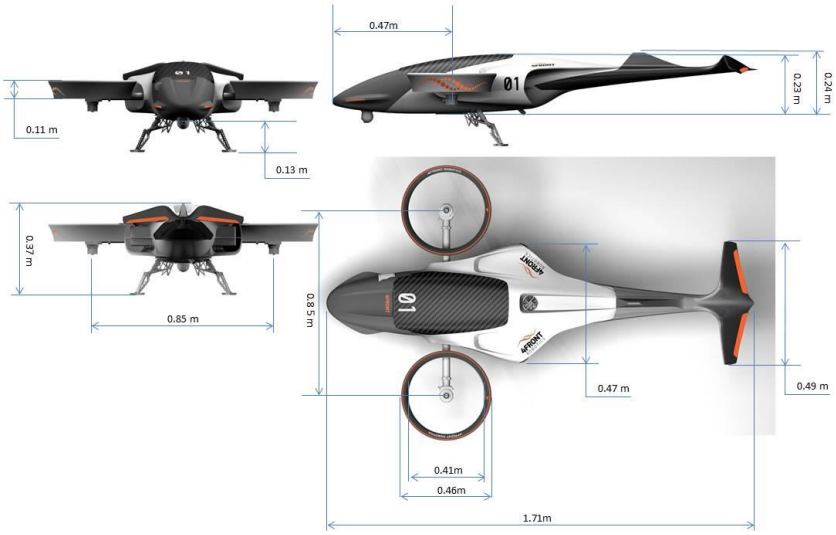
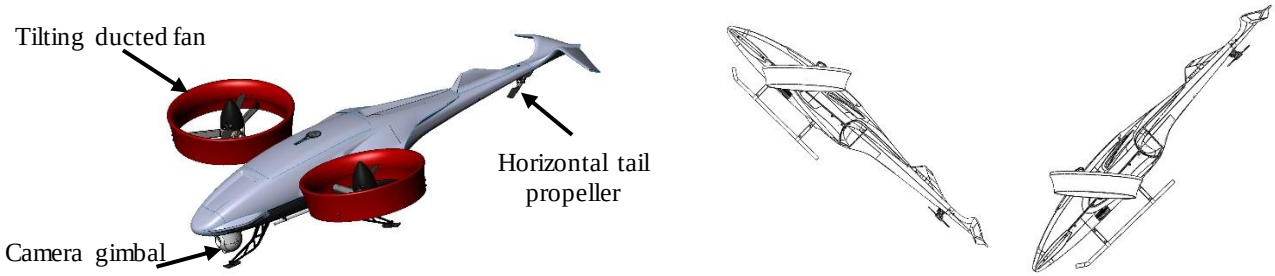


Figure 4.1: Dimensional drawing of the HM-UAV.



a) Nominal configuration b) Nose up/down pitch-hover maneuver

Figure 4.2: HM-UAV VTOL configuration.

The main purpose of the zero-lift horizontal tail propeller is to generate a pitching moment used to control/stabilize the pitch angle of the UAV during flight and not to generate

lift. As a result, the HM-UAV can take-off and land from highly sloped, prepared/unprepared terrain not possible with any other manned or unmanned aircraft.

The design of the HM-UAV allows it to be controlled (i.e., control rotations around the 3 axis and translations along the vertical and longitudinal axis of the UAV) during all flight modes (hover and high forward speed) using only the tilting twin-ducted fans and the tail propeller. The HM-UAV is unable to produce thrust in the lateral direction. The combination of duct tilting, and tail propeller provides a static pitching moment allowing the HM-UAV aircraft to remain level in hover despite any pitch imbalances (e.g., off-center loading) that can be caused by load movement of, if equipped with a robotic arm, the disturbances cause by the arm when the arm moves and manipulates objects. This allows performing pitch-hover maneuver (Figure 4.2b) despite unexpected unbalances on the aircraft. Table 4.1 summarize the methods for controlling the HM-UAV in the 6-DoF.

Table 4.1: HM-UAV controls.

Motion		Control method
Rotation	Yaw	Ducted-fans, differential tilting, fore and aft.
	Pitch	Using horizontal tail propeller collective (and combined with ducted fan tilting in case of pitching at zero velocity)
	Roll	Ducted-fans, differential collective propeller thrust
Translation	Longitudinal	Ducted-fans, coupled tilting, fore and aft
	Lateral	Coupled to roll
	Vertical	Ducted-fans, collective propeller thrust

4.1.1. HM-UAV Scalability

The HM-UAV was designed to be scalable and can be manufactured in different versions, each fitting a different range of requirements such as carrying diverse robot arm configurations.

4.1.2 Pitched Hover

The HM-UAV aircraft have the capability to hover while in a pitched attitude. This allows the aircraft to perform the following maneuvers:

- Land and takeoff from highly sloped surfaces, or rough terrain
- Land and takeoff from moving surfaces including vessels in rough sea, or on ground vehicles moving on rough terrain

where conventional aircraft would otherwise be unable to operate.

This ability also enables the HM-UAV to maneuver through smaller spaces by changing its pitch attitude relative to its direction of travel and the position & orientation of obstacles with respect to the aircraft.

Pitch hovering uses the tail rotor, and the ducts (i.e., main propellers) tilt in tandem to manipulate the pitch angle of the aircraft body, while maintaining a stable hover in one position. The ducts tilt fore and aft to maintain lift in the downward direction, while the tail rotor varies its thrust either up, or down to precisely control the pitch angle of the fuselage. When the vehicle is performing a pitch maneuver, the ducts tilt to compensate the translational tendency that the tail propeller produces when the vehicle is in a pitch angle as illustrated in Figure 4.3 and 4.4**Error! Reference source not found.**. Current allowable pitch angles for this vehicle are 85° nose up through 70° nose down.

Figure 4.5 shows the internal components of the HM-UAV which enable it to fly and perfrom pitch hover maneuvcers.

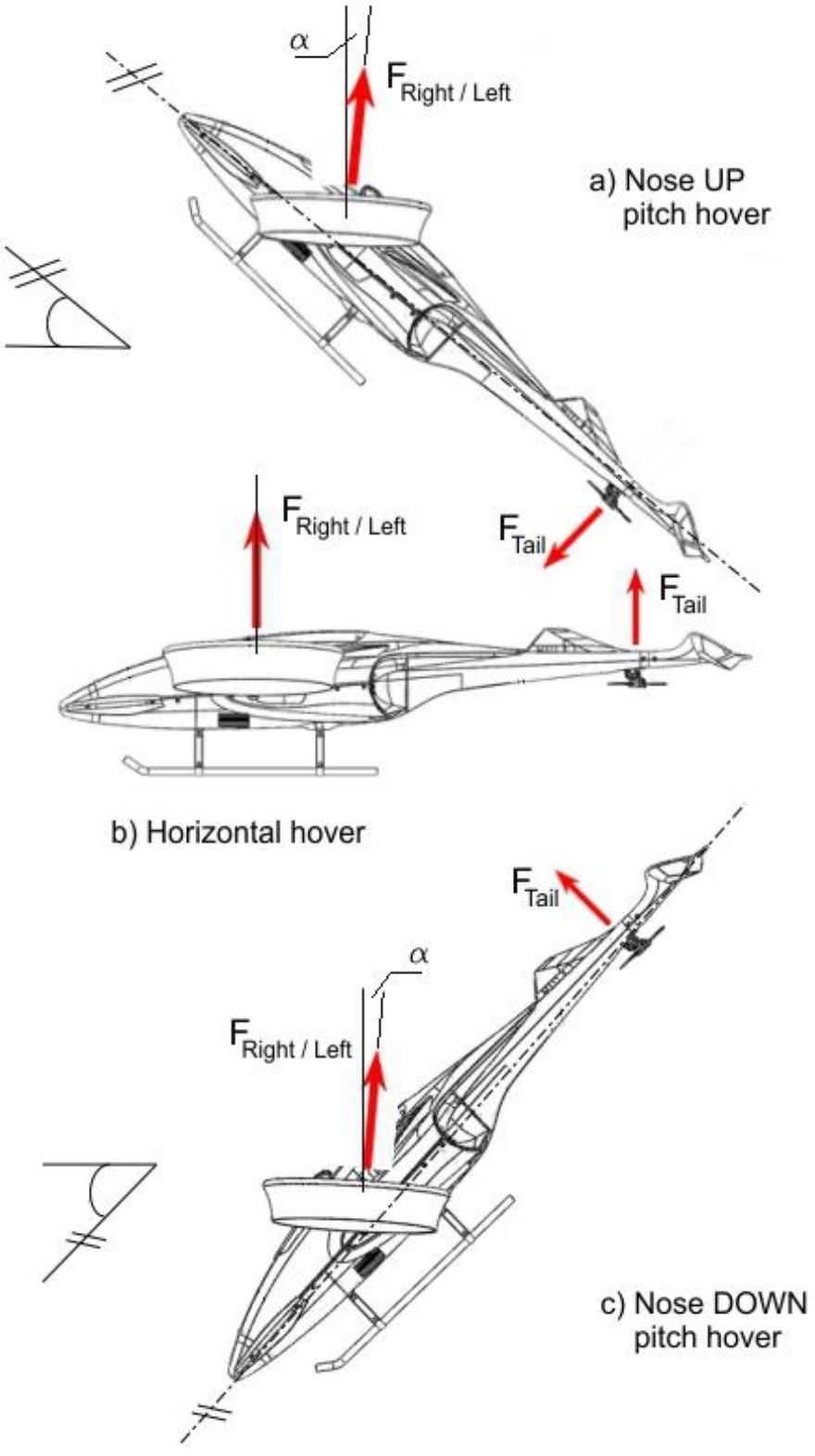


Figure 4.3: Propeller produced forces in pitch hover.

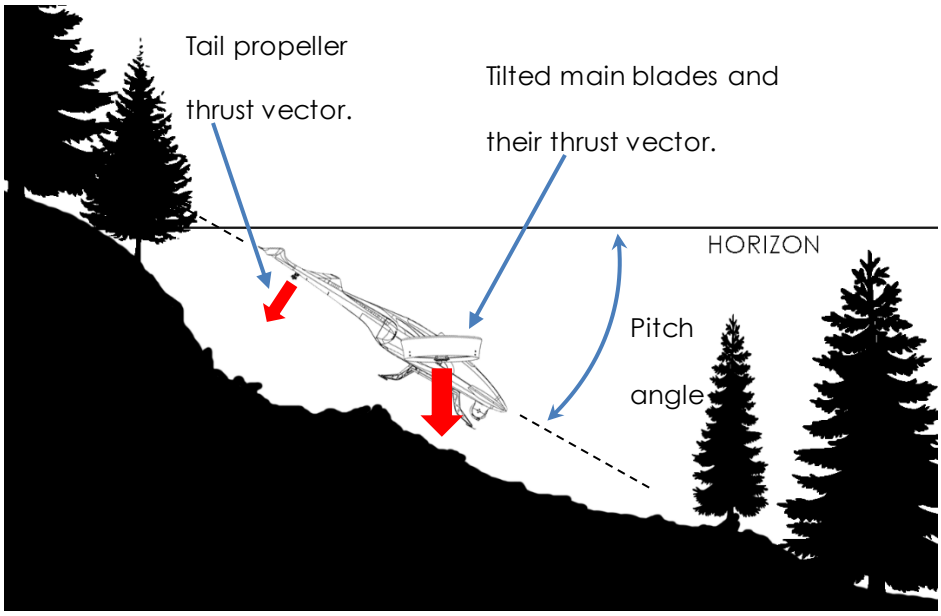


Figure 4.4: Pitch hover for slopped terrain landing.

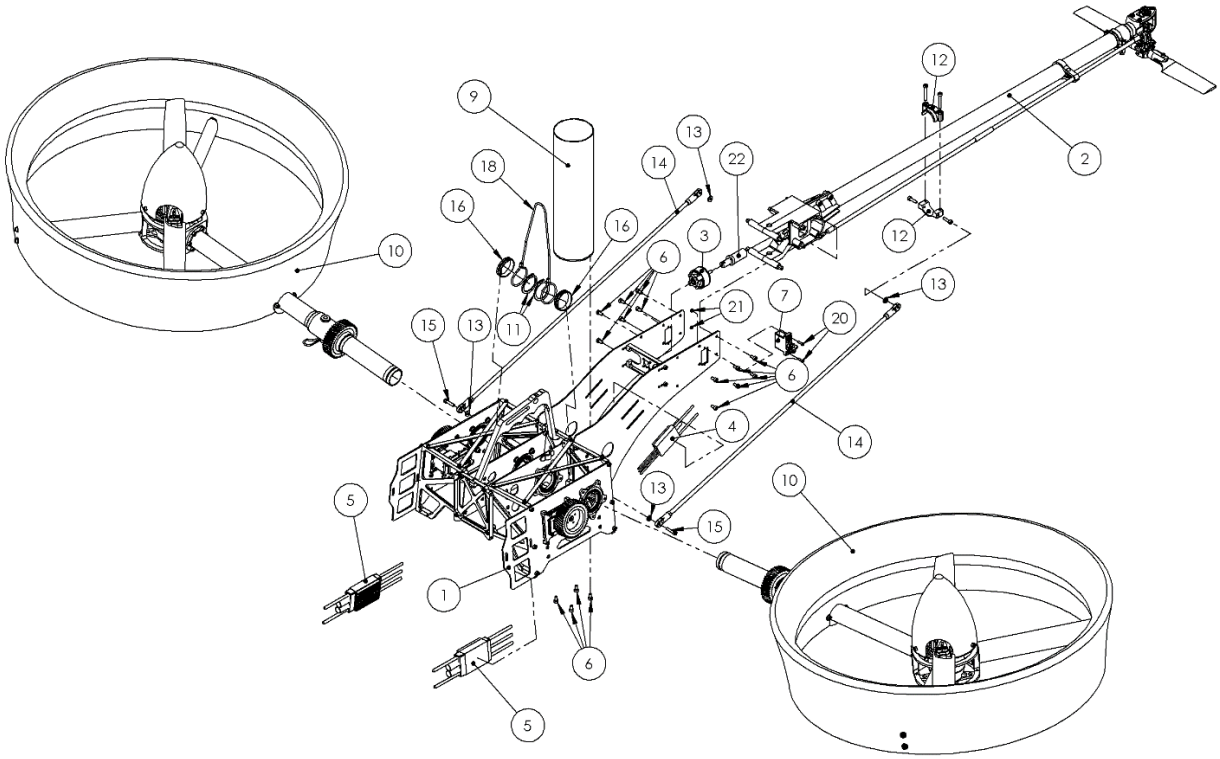


Figure 4.5: Internal Components of the HM-UAV.

4.2 Robot Arm attached to be attached to the MH-UAV

This section describes a selected robot arm used in the Unmanned Aerial Manipulator (Figure 4.1). The goal for this selection was to provide a robot arm that is light weight and has its center of mass positioned close to the base of the arm. The goal of such requirement is to minimize the disturbances (changes in the in the UAV's center of mass) to the UAV as it flies, and moves its arm to grab objects. Thus, enabling the UAV to maintain its maximum maneuverability and be able to perform pitch hover maneuvers at high angles despite being disturbed by the actions of the robot arm (situated under its fuselage).

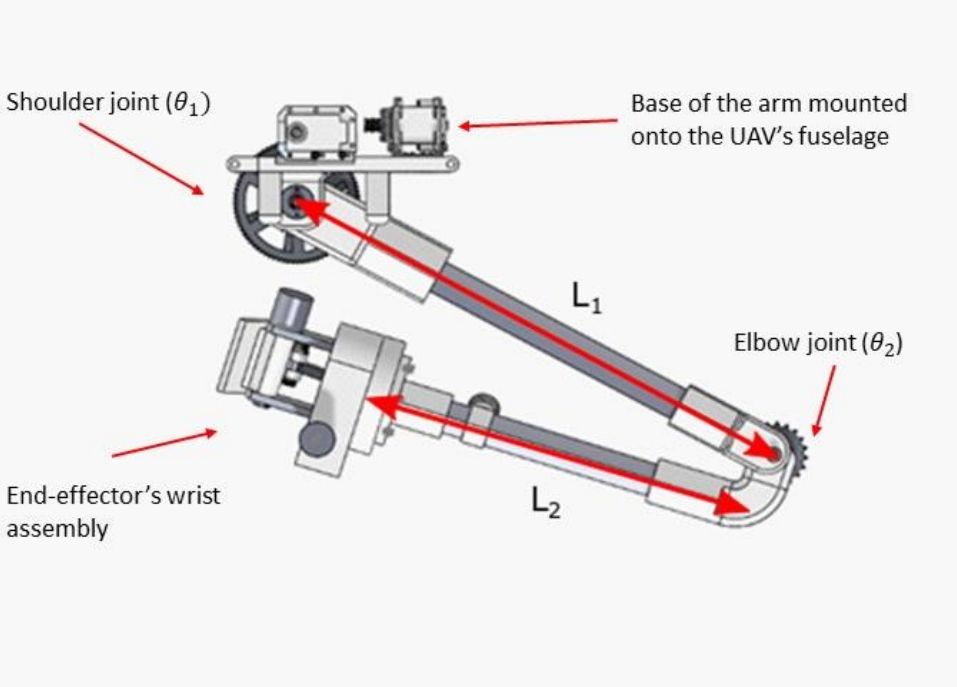


Figure 4.6: Full 2-DOF arm with attached robotic wrist.

The proposed robotic arm is a revolute-revolute 2-DOF robot arm designed using a light-weight structure (0.425kg). The lightweight design is achieved with large diameter thin wall carbon fiber links, and ABS plastic joints. The joints use nylon bushings with shoulder bolts for

the axel. The ABS joints can be 3D printed in small scale production or cast/injection molded when manufactured at larger scale. The arm is mounted under the HM-UAV fuselage and constraint to only move along the longitudinal plane of the UAV (Figure 3.1). Thus, the arm's shoulder represents an arm that can only move in a plane (i.e., along the longitudinal plane of the UAV). The rotation of the arm's shoulder to enable it to move in 3 dimensions is provided by the yaw motion of the UAV unit. The reason for this was to prevent the arm to cause a roll motion on the UAV in cases when the arm was rotated around its shoulder. The arm specifications are shown in Table 4.1.

Table 4.2: Robot Arm Specifications.

Max Capacity of Arm (Kg)	1.5
Mass of Arm (Kg)	0.425
Max Moment of Inertia of Arm ($Kg\ m^2$)	0.1017
Mass of Motors (Kg)	0. 144
Max range of tip of arm (m)	0.8
Max Motor Speed (rpm)	50
Max Motor Torque (Nm)	2.5

The design of the arm was specifically made to reduce the motion of its CoM and thus minimize any disturbances that it could produce on the UAV. The following plots (Figures 4.7, and 4.8) illustrate the movement of the CoM of the robotic arm with varying masses at the end effector. Figure 4.7 represents the area within which the CoM of the arm moves (black region) as well as the workspace of the robotic arm (light blue region) when the arm carries no load (i.e., 0 kg at its end-effector). Figure 4.8 represent the area within which the CoM of the arm moves

(black region) when the arm carries a load of 2 lbs. (0.909 kg) at its end-effector. The light and darker black regions on Fig. 4.8 are due to the number of data points used to plot the figure, but clearly exemplify how much the arm disturbances on the UAV could increase when carrying varying loads.

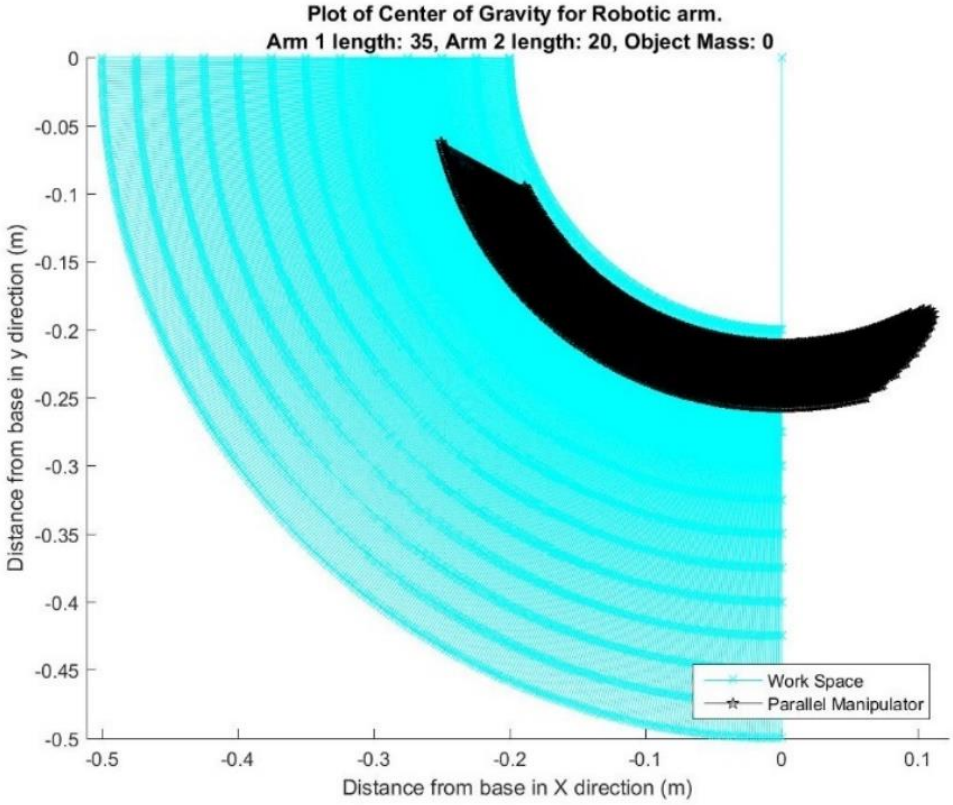


Figure 4.7: Movement of Arm's Center of Mass with no load.

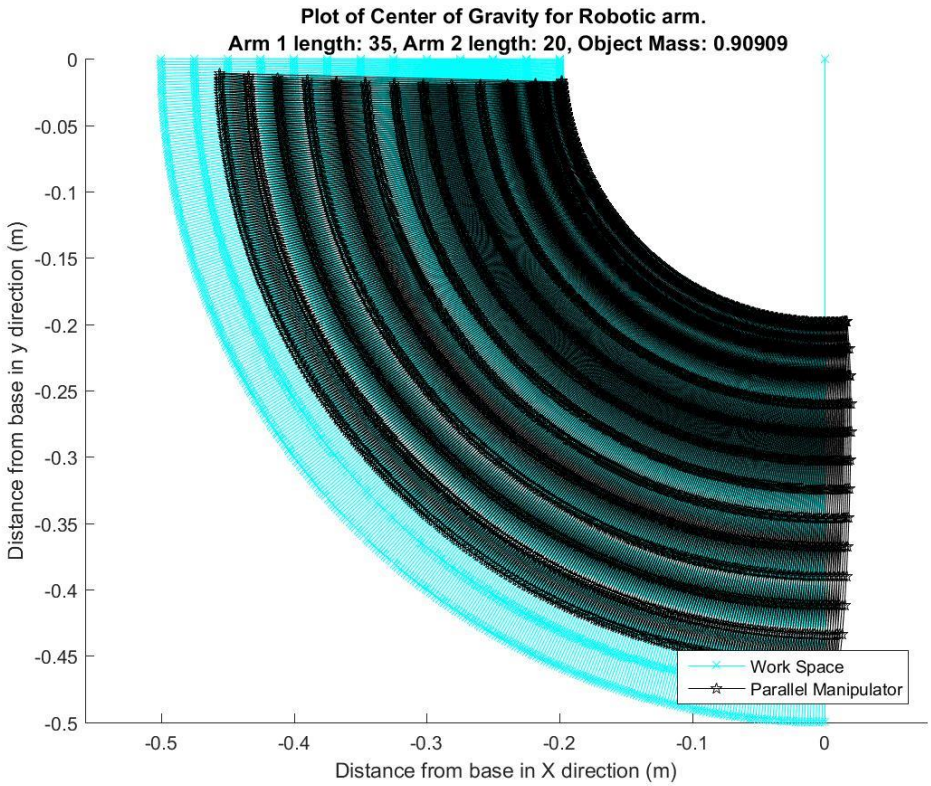
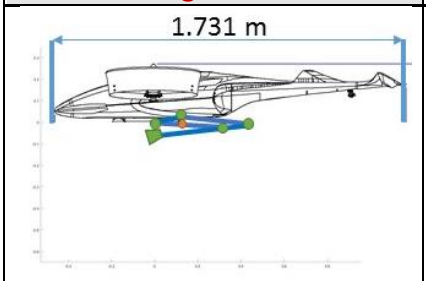
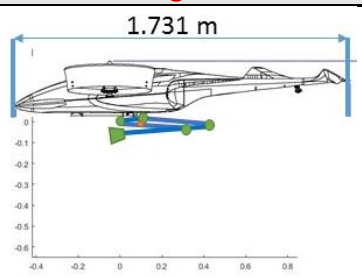
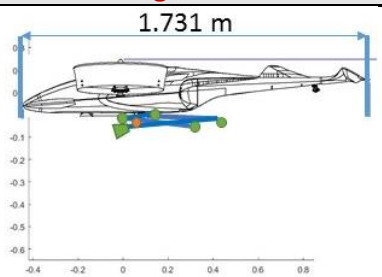
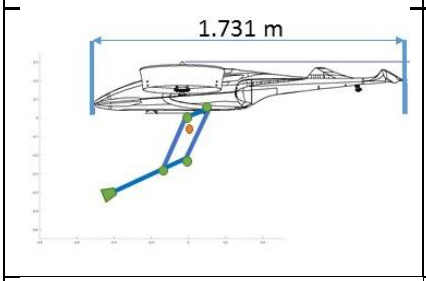
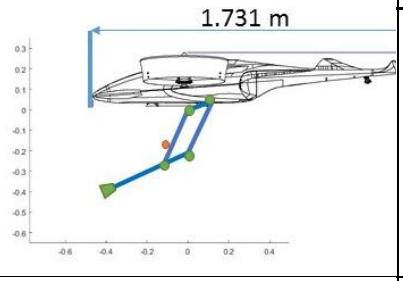
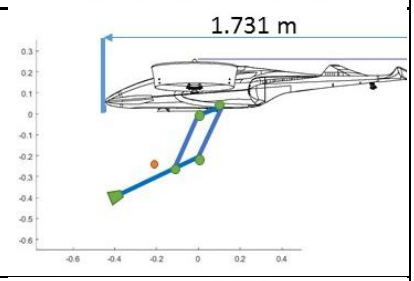
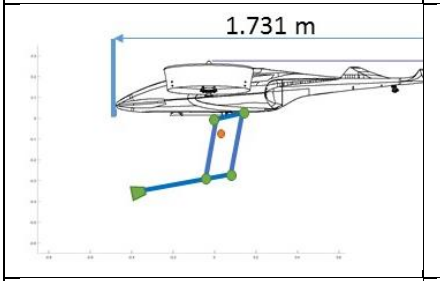
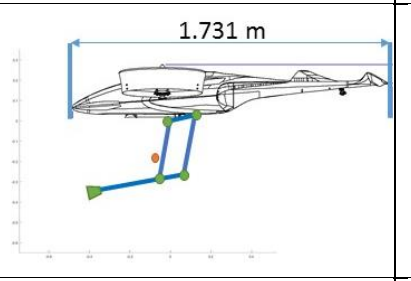
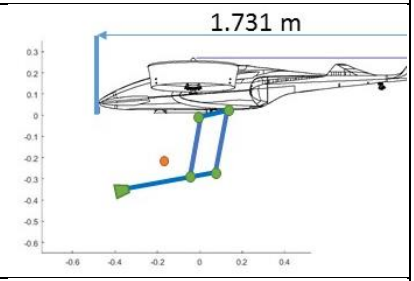
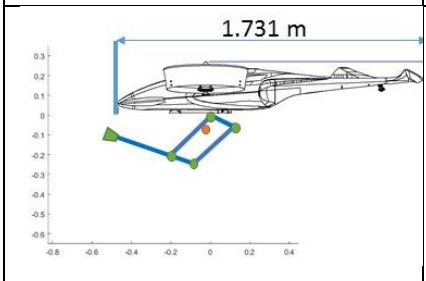
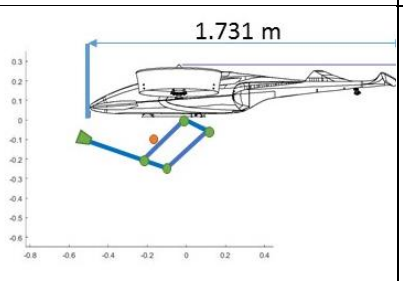
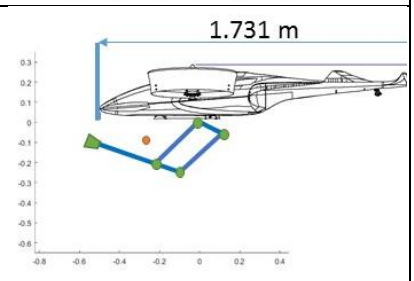
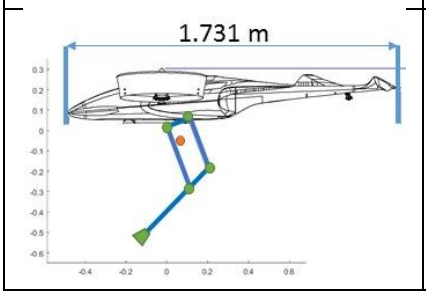
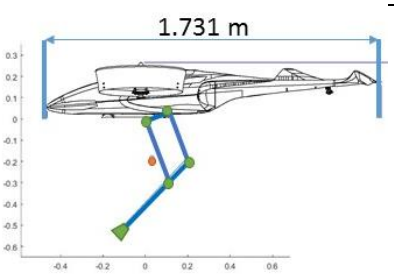
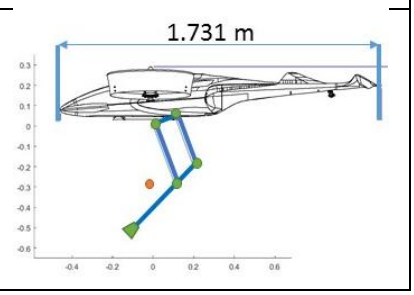


Figure 4.8: Movement of Arm's Center of Mass with 2 Ib load

Table 4.2 illustrate some of the data points (orange dots) used to generate the CoM region in Figures 4.7 and 4.8. The first column of Table 4.2 illustrates the position of the robot arm's center of mass at a number of specific robot arm positions where the arm does not have any load at the tip of the end-effector. Please note that the arm does not include a robotic gripper and that a slightly different arm design is used in the figures for illustrative purposes. However, the in movement of the CoM of the arm for various end-effector positions should be similar to that of the employed revolute-revolute arm in this work. The location of the robot arm's center of mass in Table 4.2 is represented by an orange coloured dot.

Table 4.3: Movement of the arm's CoM during its motion.

Arm’s center of gravity with **0.0 kg load**.	Arm’s center of gravity with **0.25 kg load**.	Arm’s center of gravity with **1 kg load**.
		
		
		
		
		

The workspace of the robot arm being positioned under the fuselage of the HM-UAV, used in this work, is illustrated in Figure 4.9, where numerous arm positions are included with the corresponding Joint 1 (θ_1) and Joint 2 (θ_2) angles.

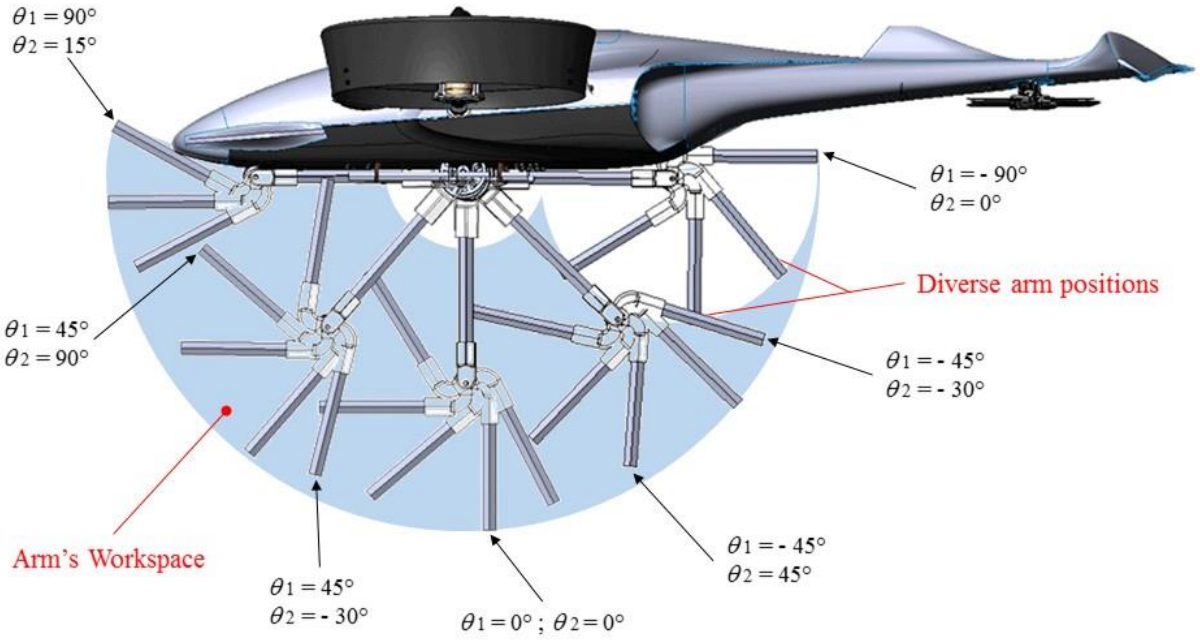


Figure 4.9: Workspace of the robotic arm with various arm positions.

Chapter 5 - Kinematic Modeling

To perform the control and control design of a UAM a suitable mathematical representation is needed. In this chapter, the complete kinematic model of the HM-UAV having a pitch trim horizontal tail rotor and an attached 2 DOF robotic arm (Figure 5.1) used in this research is presented.

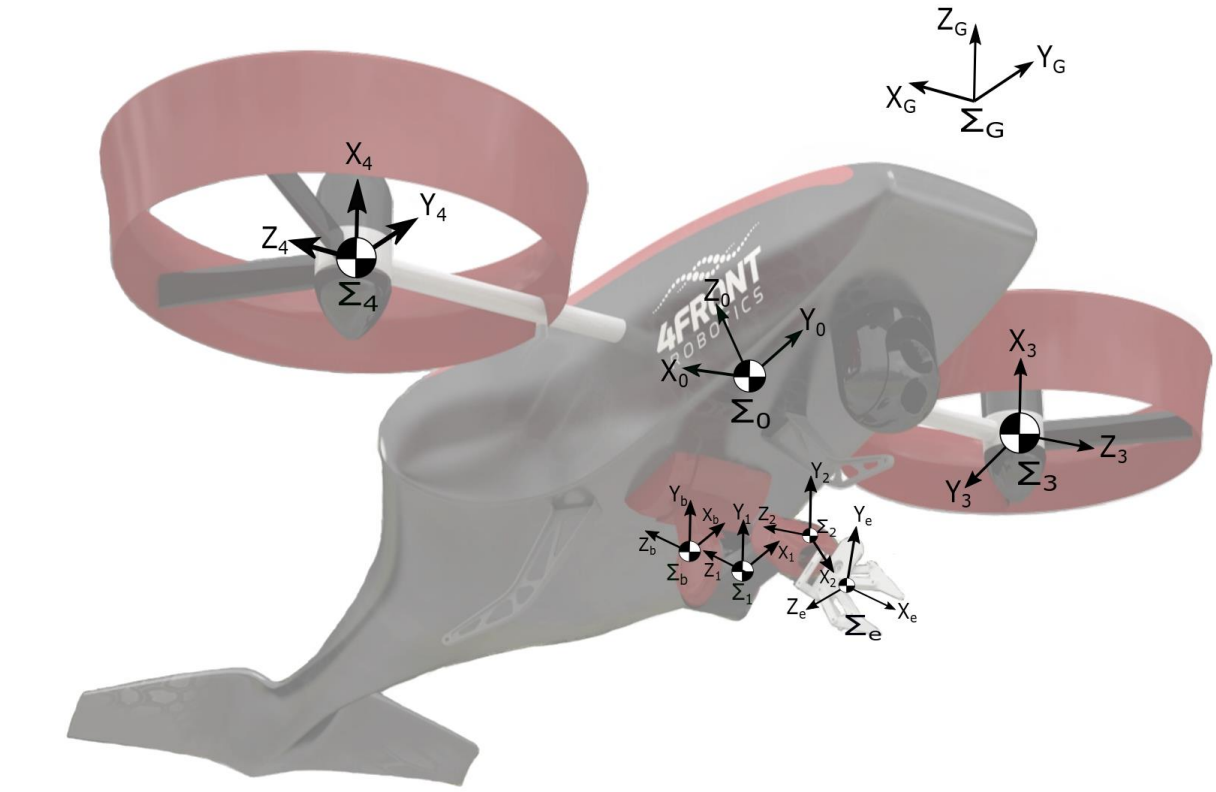


Figure 5.1: Global Coordinate frame and frames of reference for the links in the UAS.

5.1 Introduction

The forwards kinematic relationship between the HM-UAV and the manipulator is formulated for the purpose of understanding how the end-effector will need to move based on the current state of the joint inputs and states of the HM-UAM which can then be used for control

purposes. To do this, first the orientation of important frames of references for the HM-UAM, such as the global or inertial frame (Σ_G), base frame (Σ_0), and end-effector frame (Σ_e), and their relationship with the other vehicle frames is defined. Next, the position of the important system linkages such as the rotational joints of the HM-UAV and the arm are defined. Using the developed transformations between the position and orientation of the frames of reference, the position of the links comprising the HM-UAM can be expressed in any desired frame of reference. The velocity kinematics are also formulated which is split into two parts: linear and angular velocities. The linear and angular velocity of the end-effector w.r.t the HM-UAV is then formulated using the velocity Jacobian of the manipulator. The velocity state of the HM-UAV (angular and linear) is subsequently formulated from sensor information and the transformations between the corresponding frames of references.

5.2 HM-UAM Position and Orientation: Forward Kinematics.

In this section, the positioning and orientation of the frames of reference in the HM-UAV are defined. The components comprising/forming the aerial platform used in this research are assumed to be rigid. As detailed in Chapter 4, this system is comprised of a rigid UAV chassis with two main rotor pods that tilt about their joint axis, and a variable horizontal pitch tail rotor attached to the rigid frame of the UAV as shown in Figures 5.1, and 5.2.

The frames of references used to derive the kinematic model of the UAV are shown in Figure 5.1 where frame of reference Σ_0 is used to denote the vehicle's body frame of reference positioned at the CoM (Centre of Mass) of the UAV. This frame of reference is aligned with the center axis of symmetry of the UAV.

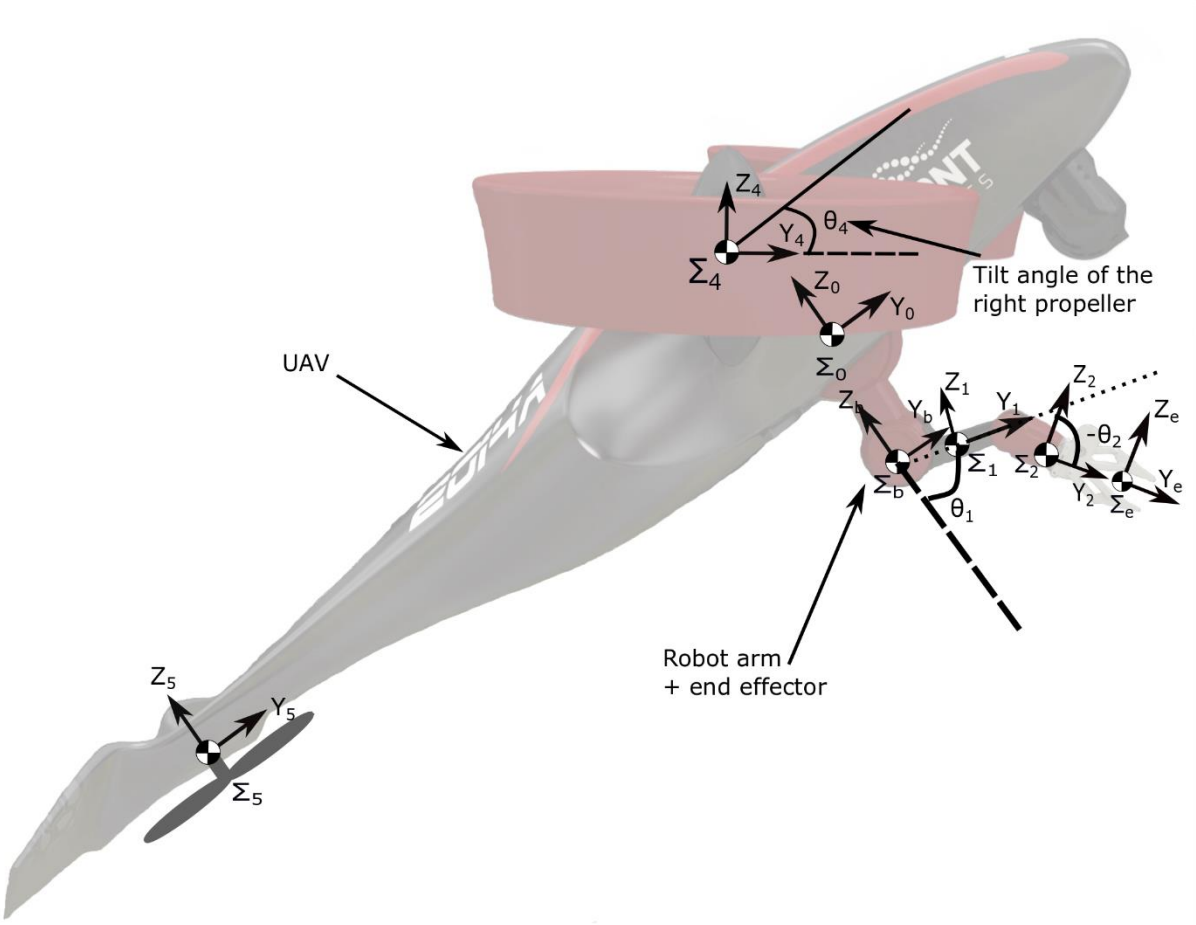


Figure 5.2: Coordinate frames of reference for the HM-UAV having a horizontal pitch trim tail rotor and the attached robotic arm.

Its Y_0 axis is pointing in the forwards direction of the UAV, the X_0 axis is directed in the lateral direction and the Z_0 axis is pointing up towards the top of the aircraft's fuselage. As a result, the coordinates (x_G, y_G, z_G) w.r.t the inertial frame, Σ_G, define the position of the HM-UAV. The UAV is positioned in a global or inertial frame Σ_G which is used to represent the vehicle's position and orientation in the real world. The x_G, y_G, z_G position of the UAV in the global frame is given by the (3x1) vector $p_G^0 = [\, x_G \; y_G \; z_G]^T$ (shown in Figure 5.1), and its orientation is given by the rotation matrix R_0^G (Equation 5.1) which is defined based on the quaternion $a + bi + cj +$

dk of the UAV, where $a, b, c,$ and d are the fundamental quaternion units defining a hypercomplex set of numbers with one real and three imaginary parts [49].

$$R_0^G = \begin{bmatrix} a^2 + b^2 - c^2 - d^2 & 2(bc + ad) & 2(bd - a\theta_2) \\ 2(bc - ad) & a^2 - b^2 + c^2 - d^2 & 2(cd + ab) \\ 2(bd + ac) & 2(cd - ab) & a - b - c + d \end{bmatrix} \tag{5.1}$$

The end-effector's frame of reference (Figure 5.1) is denoted as Σ_e.The frame of reference for each link i comprising the arm is defined as Σ_i and is centered at the center of mass (CoM) of each link i (see Figure 5.2) for $i = 1, 2$ for the specific robot arm.

The orientation of the UAV represented by $\vec{\Phi}_0$ w.r.t the inertial frame Σ_G is given by the Tait-Bryan XYZ Roll-Pitch-Yaw angles α, β, γ (roll, pitch, yaw) given the following formulation of the rotation matrix R_B^G:

$$R_0^G = \begin{bmatrix} c_\beta c_\gamma & -c_\beta s_\gamma & s_\beta \\ c_\alpha s_\gamma + c_\gamma s_\alpha s_\beta & c_\alpha c_\gamma - s_\alpha s_\beta s_\gamma & -c_\beta s_\alpha \\ s_\alpha s_\gamma - c_\alpha c_\gamma s_\beta & c_\beta s_\alpha + c_\alpha s_\beta s_\gamma & c_\alpha c_\beta \end{bmatrix} \tag{5.2}$$

where c_i and s_i are the cosine and sine of angle i, respectively. The position of the end effector p_e^G w.r.t the frame of reference Σ_G is given by:

$$p_e^G = p_0^G + R_0^G \, p_e^0 \tag{5.3}$$

where the vector p_e^0 describes the position of the end-effector w.r.t the base frame Σ_0, and p_0^G is the position of the UAV's frame Σ_0 w.r.t the global frame Σ_G. In general terms the position of the frame of reference Σ_i with respect to Σ_G, p_i^G, is given by:

$$p_i^G = p_0^G + R_0^G \, p_i^0 \tag{5.4}$$

where the vector p_i^0 describe the position of Σ_i w.r.t the base coordinate frame Σ_0. As a result, the position of the end-effector of the manipulator w.r.t the frame Σ_0 can be expressed as:

$$p_e^0 = d_1^G + \begin{bmatrix} 0 \\ L_1 s_1 + L_2 s_{12} \\ L_1 c_1 + L_2 c_{12} \end{bmatrix} \tag{5.5}$$

where s_{12}, and c_{12} are used to describe the functions $\sin(\theta_1 + \theta_2)$, and $\cos(\theta_1 + \theta_2)$ respectively, and d_1^G is the position of the base of the manipulator w.r.t. the CoM of the UAV resolved in Σ_0. The orientation of the end-effector's frame of reference Σ_e w.r.t the global frame Σ_G is described by the following rotation matrix:

$$R_e^G = R_0^G R_e^0 \tag{5.6}$$

where the rotation matrix R_e^0 describes the orientation of the end-effector Σ_e w.r.t the body frame Σ_0, R_e^0 is the rotation matrix based on the rotation matrices R_1^0, and R_2^1 which are based on the position of Joints 1 and 2 of the manipulator as follows:

$$R_1^0 = \begin{bmatrix} 0 & \sin\theta_1 & -\cos\theta_1 \\ 0 & \cos\theta_1 & \sin\theta_1 \\ -1 & 0 & 0 \end{bmatrix} \tag{5.7}$$

$$R_2^1 = \begin{bmatrix} \cos\theta_2 & \sin\theta_2 & 0 \\ -\sin\theta_2 & \cos\theta_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.8}$$

The rotation matrix defining the orientation of the end-effector w.r.t ink 2 R_e^2 is written as:

$$R_e^2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.9}$$

The rotation matrix defining the orientation of the end-effector w.r.t the base frame R_e^0 is written as:

$$R_e^0 = R_e^2 R_2^1 R_1^0 \tag{5.10}$$

where θ_1 and θ_2 define the joint angles for the proximal and distal linkages of the robot arm as illustrated in Figure 5.2. The forwards kinematics of the manipulator is determined through the known vehicle and joint space inputs $q = [x_G\ y_G\ z_G\ \alpha\ \beta\ \gamma\ \theta_1\ \theta_2]^T$ from where one can determine the resulting state of the end-effector. There are three position coordinates for the end-effector denoted as $X_e = [\ x_e\ y_e\ z_e]^T$ w.r.t Σ_0.

The other important link positions in the HM-UAM that need to be computed w.r.t the base frame Σ_0 and the global frame Σ_G are the positions and rotations of the two main rotors Σ_3, Σ_4 of the HM-UAV and the position of the base of the robot arm Σ_b. As a result, the z_i axis of the rotor frame is aligned with the axis of rotation of the rotors. The rotation matrices R_i^0 for $i =$ 3 and 4 can be expressed as:

$$R_i^0 =$$

$$\begin{bmatrix} cos\ q_i + u_{x,i}^2(1 - cos\ q_i) & u_{x,i}u_{y,i}^2\left(1 - cos\ q_i\right) + u_{z,i}\ sin\ q_i & u_{x,i}^2u_{y,i}^2\left(1 - cos\ q_i\right) + u_{z,i}\ sin\ q_i \\ u_{x,i}u_{y,i}\left(1 - cos\ q_i\right) - u_{z,i}\ sin\ q_i & cos\ q_i + u_{y,i}^2\left(1 - cos\ q_i\right) & u_{y,i}u_{z,i}\left(1 - cos\ q_i\right) + u_{x,i}\ sin\left(q_i\right) \\ u_{x,i}u_{z,i}\left(1 - cos\ q_i\right) + u_{y,i}\ sin\ q_i & u_{y,i}u_{z,i}\left(1 - cos\ q_i\right) - u_{x,i}\ sin\ q_i & cos\ q_i + u_{z,i}^2(1 - cos\ q_i) \end{bmatrix} \tag{5.11}$$

where $u_i = [u_{x,i} \quad u_{y,i} \quad u_{z,i}]^T$ is the axis of rotation of joint i defined in Σ_i, and q_i is the corresponding joint position shown in Figure 5.3 and defined in Table 5.1. The position of rotor i, p_i^0 defined in Σ_0 can be found using eqn 5.12.

$$p_i^0 = R_i^0 p_i + d_i \tag{5.12}$$

where p_i is the vector from $\Sigma_{(i-1)}$ to u_i in the $\Sigma_{(i-1)}$ frame of reference, and d_i is the vector from u_i to Σ_i in the Σ_i frame of reference as illustrated in Figure 5.4.

Table 5.1: UAS State Variables

GENERALIZED COORDINATE NUMBER	**DESCRIPTION**
$\boldsymbol{q_1}$	X position of UAV in Σ_0 (x_0)
$\boldsymbol{q_2}$	Y position of UAV in Σ_0 (y_0)
$\boldsymbol{q_3}$	Z position of UAV in Σ_0 (z_0)
$\boldsymbol{q_4}$	Pitch (β)
$\boldsymbol{q_5}$	Yaw (γ)
$\boldsymbol{q_6}$	Roll (α)
$\boldsymbol{q_7}$	Position of Joint 1 of Arm (θ_1)
$\boldsymbol{q_8}$	Position of Joint 2 of Arm (θ_2)

The position of the frame of reference for the base of the attached robotic arm Σ_b has been defined to have the same orientation as the MM-UAV frame Σ_0 (Figure 5.2) and its position is given by the vector p_b.

$$p_b^0 = p_b \tag{5.13}$$

Equations 5.10, 5.11, and 5.12 define the complete forward kinematics of the HM-UAM and can be used to determine the pose of every component comprising the UAM system.

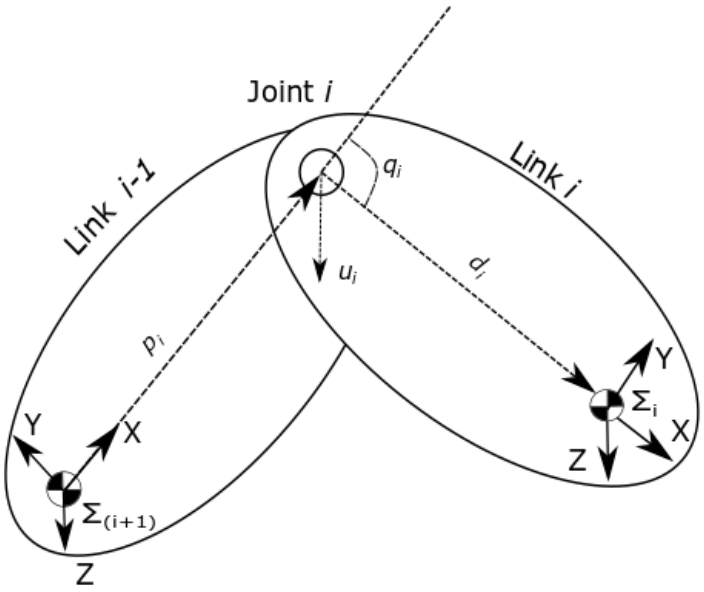


Figure 5.3: Depiction of link relations.

5.3 HM-UAM Forwards Velocity Kinematics.

With the kinematic equations developed in Section 5.2 it is possible to proceed to obtain both components of velocity for all key links comprising the HM-UAM. The linear velocity of the end-effector in the global frame $\dot{p}_e^G$ can be found by differentiating Eqn (5.3) resulting in Eqn 5.14:

$$\dot{p}_e^G = \dot{p}_0^G - Skew(R_0 p_e^0)\omega_0 + R_0 \dot{p}_e^0 \tag{5.14}$$

where $Skew(.)$ is the (3x3) skew-symmetric matrix operator, and ω_0 is the angular velocity of the HM-UAV in Σ_G. The velocity of end-effector w.r.t the base $\dot{p}_e^0$ can be found from the linear velocity Jacobian of the manipulator written as:

$$\dot{p}_e^0 = J_{v_e}^0 \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} \tag{5.15}$$

where $J_{v_e}^0$ is the linear velocity Jacobian. The linear velocity Jacobian $J_{v_e}^0$ is in turn expressed as:

$$J_{v}^{0}{}_{e} = \begin{bmatrix} 0 & 0 \\ L_1c_1 + L_2c_{12} & L_2c_{12} \\ -L_1s_1 - L_2s_{12} & -L_2s_{12} \end{bmatrix} \tag{5.16}$$

where L_1 and L_2 are the manipulator's length links 1 and 2 respectively (Figure 4.6). The angular velocity ω_e^G of the end-effector expressed in global coordinates Σ_G is written as:

$$\omega_e^G = \omega_0^G + R_e^0\omega_e^0 \tag{5.17}$$

where ω_e^0 is the angular velocity of the end-effector relative to the coordinate frame Σ_0. The angular velocity of the end-effector frame ω_e^0 as a function of the manipulator's joint velocity is thus expressed as:

$$\omega_e^0 = J_{\omega}^{0}{}_{e} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} \tag{5.18}$$

where $J_{\omega_e}^0$ is the angular velocity Jacobian of the manipulator in base coordinates Σ_0. The angular velocity Jacobian of the manipulator $J_{\omega_e}^0$ is defined as:

$$J_{\omega}^{0}{}_{e} = \begin{bmatrix} 1 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{5.19}$$

Equation 5.19 shows that ω_e^0 is only a function of $\dot{\theta}_1$ and $\dot{\theta}_2$ and only about the x_0 axis. The (6 x 1) vector of the generalized velocity of the end-effector v_e^0 defined in the base coordinate frame Σ_0 is expressed as:

$$v_e^0 = \begin{bmatrix} \dot{p}_e^0 \\ \omega_e^0 \end{bmatrix} \tag{5.20}$$

As a result, the generalized velocity represented in Eqn 5.20 can be expressed in terms of the generalized manipulator Jacobian, J_e^0, such that:

$$v_e^0 = J_e^0 \dot{\theta} \tag{5.21}$$

where $\dot{\theta} = [\dot{\theta}_1 \quad \dot{\theta}_2]^T$ is the (2 x 1) joint velocity vector of the manipulator and the generalized manipulator Jacobian is expressed as:

$$J_e^0 = \begin{bmatrix} J_{v_e}^0 \\ J_{\omega_e}^0 \end{bmatrix} \tag{5.22}$$

Thus, the generalized velocity vector of the end-effector v_e^G expressed in global coordinates Σ_G is written as:

$$\vec{v}_e^G = \begin{bmatrix} \dot{p}_e^G \\ \omega_e^G \end{bmatrix} = J_0 v_0^G + \begin{bmatrix} R_0^G & 0_3 \\ 0_3 & R_0^G \end{bmatrix} J_e^0 \dot{\theta} \tag{5.23}$$

where v_0^G is the generalized velocity of the base frame (the HM-UAV frame) w.r.t the global frame. The parameter, 0_3 is used to represents a (3 x 3) null matrix, and J_0 is the generalized velocity Jacobian for the base coordinate frame (Equation 5.25). The generalized velocity of the HM-UAV frame is expressed as:

$$v_0^G = \begin{bmatrix} \dot{p}_0^G \\ \omega_0^G \end{bmatrix} \tag{5.24}$$

and the generalized velocity Jacobian for the base coordinate frame is then expressed as:

$$J_0 = \begin{bmatrix} I_3 & -Skew(R_0^G p_e^0) \\ 0_3 & I_3 \end{bmatrix} \tag{5.25}$$

where I_3 is a (3 x 3) identity matrix. The inputs for the velocity Jacobian of the manipulator are the six velocity states of the HM-UAV v_0^G (Equation 5.24) and shown again in its extended form in Equation 5.26.

$$v_0^G = \begin{bmatrix} \dot{x}_0^G \\ \dot{y}_0^G \\ \dot{z}_0^G \\ \omega_{x_0}^G \\ \omega_{y_0}^G \\ \omega_{z_0}^G \end{bmatrix} \tag{5.26}$$

The rate of change in orientation of the UAV $\Phi_0 = [\dot{\alpha}\ \dot{\beta}\ \dot{\gamma}]$ is not the same as the angular velocity ω_0 of the HM-UAV ($\Phi_0 \neq \omega_0$) but the relationship between $\dot{\phi}_0$ and ω_0 need to be used. The transformation from Φ_0 to ω_0 can be expressed as:

$$\begin{bmatrix} \omega_{x_0} \\ \omega_{y_0} \\ \omega_{z_0} \end{bmatrix} = \begin{bmatrix} \cos\alpha\cos\gamma & \sin\gamma & 0 \\ -\cos\beta\sin\gamma & \cos\gamma & 0 \\ \sin\beta & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix} \tag{5.27}$$

Equations 5.22, 5.23 and 5.26 define the complete forward velocity kinematics of the HM-UAM and can be used to compute the velocity of every component comprising the UAM system.

5.3 Conclusion

In this chapter, the mathematical formulation of the forwards kinematics for the HM-UAM is presented. The formulation includes the position and velocity of the end-effector as a function of the HM-UAV and the joint space inputs q and $\dot{q} = [\dot{x}_0\ \dot{y}_0\ \dot{z}_0\ \dot{\alpha}\ \dot{\beta}\ \dot{\gamma}\ \dot{\theta}_1 \dot{\theta}_2\ ]$. The transformation between the frames of references of the links comprising the HM-UAM were also defined in this chapter. This included their relative orientation and position to both the base and global frames of reference.

Chapter 6 - Dynamic Modeling

To develop the control mechanism of a HM-UAM, the mathematical modeling of the dynamics of the UAM which describe how the state of the system changes in response to inputs is required. The state of the HM-UAM defined in terms of the Σ_0 frame of reference is given by the vector $\zeta = \left[X_0^T \ \Phi_0^T \ \theta^T \ \dot{X}_0^T \ \Phi_0^T \ \dot{\theta}^T \right]^T$ where $X_0 = [x_0 \quad y_0 \quad z_0]$ is the position vector of the HM-UAV w.r.t the frame Σ_G, $\Phi_0 = [\alpha \ \beta \ \gamma]^{\mathrm{T}}$ is the orientation of the HM-UAV w.r.t frame Σ_G, and $\theta = [\theta_1 \ \theta_2 \ \theta_3 \ \theta_4]^T$ is the vector of joint positions (tilting propellers θ_3 and θ_4, and robot arm joints θ_1 and θ_2), and their time derivatives $\dot{X}_0, \dot{\Phi}_0, \dot{\theta}$ thus, $\zeta = \left[x_0 \ y_0 \ z_0 \ \alpha \ \beta \ \gamma \ \theta_1 \ldots \theta_4 \ \dot{x}_0 \ \dot{y}_0 \ \dot{z}_0 \ \dot{\alpha} \ \dot{\beta} \ \dot{\gamma} \ \dot{\theta}_1 \ldots \ \dot{\theta}_4 \right]$. Herein, the dynamic modeling of the UAV + Arm is derived by applying the newly developed recursive Newton-Euler multibody dynamics (RNEMD) algorithm from [49]. The multibody dynamics algorithm from [49] developed in 2016 was chosen because of the advantages in speed of computation it provides. To increase processing speed, the algorithm pre-defines specific terms (e.g., $H_j = \hat{I}_j^A \ \hat{c}_j + \hat{Z}_j^A$) which have no clear physical definition or interpretation but once obtained, such terms assist in performing the required operations in the algorithm (e.g., computing the total Coriolis force times the Inertia for a given link comprising the system) faster. The required terms used in the algorithm (i.e., C_i, B_i, U_i, D_i, H_i and K_i) will be described and their mathematical formulations defined in Section 6.3 [49]. In what follows the reference transformations, as well as the kinematic relationships between the links of the UAS derived in Chapter 5 are used. The proposed multibody-dynamics algorithm for the UAS contains multiple passes for each time step. For the first pass the current state of the UAS is calculated starting with the base link and moving from link $i = 1$ to the final link $i = n$. In this pass, the relative pose of each of the links comprising the HM-UAM is

calculated as well as their velocity and acceleration. The second pass within the RNEMD algorithm calculates the generalized force acting on system starting with the last link $i = n$ and ending with the base link. Section 6.1 sets up the dynamic calculations, and the parameters used. The calculations for the first pass are presented in Section 6.2. The following section (Section 6.3) describes the second pass of the multibody-dynamics algorithm where the inertial calculations and the external forces acting on the links are calculated. The resulting generalized force acting on each of the links are calculated next. The multibody dynamics algorithm basically amounts to solving Eqn 6.1 for $\hat{F}_i^I$ which is the generalized force acting on link i. The generalized force acting on each link in the robot arm is then projected onto its respective joint axis to find the torque that the arm joint needs to overcome to execute the desired task/motion.

6.1 Dynamic Modeling Set-up

The recursive Newton-Euler multibody dynamics method divides the multibody system into a series of links, with a single base link and a set of n non-base links. For the proposed HM-UAM, the HM-UAV chassis is considered as the base link with four non-base links which are the two-manipulator links, and the two-main tilting rotor pods. The links are all connected to each other through revolute joints. Euler's two laws of motion are solved using a recursive method for each link i and the base link 0. Typically, the base link for traditional robotic arm systems is fixed, however since the base link in the HM-UAM system it is the UAV, the base link is movable. Thus, the generalized forces acting on each link can be computed via Equation 6.1.

$$\hat{F}_i^I = \hat{I}_i^A \hat{a}_i + \hat{Z}_i^A \tag{6.1}$$

Where in Eqn 1, $\hat{F}_i^I$ is the generalized force acting on link i from within the system, $\hat{I}_i^A$ is the generalized inertia of the link, $\hat{a}_i$ is the generalized acceleration the link experiences, and $\hat{Z}_i^A$ is the effective generalized force acting on link i due to the motion of the system. For this recursive

multibody formulation [49], a method of solving Eqn 6.1 is needed. To achieve this the links comprising the HM-UAM are set-up and their dynamic relationships determined. The set up of the links and their dynamic relationship can be shown using a tree structured diagram. The set up of the HM-UAM which includes the base link, parent and child links is shown in Figure 6.1. The base link is the parent link of all links that directly attached to it (i.e., Link 1, 3, and 4). Links that descend from other non-base links are considered the child links and the link directly above them in the chain can be considered their parent link. For example, from Figure 6.1 Link 2 is the child link of Link 1, and Link 1 is the parent link of Link 2. The set of child link(s) that a link i has is denoted as $\mu(i)$ and $i \in \mu(i)$. Links that precede the other links may not necessarily be their parent link. For example, in Figure 6.1, the parent link of Link 3 is Link 0, not Link 2.

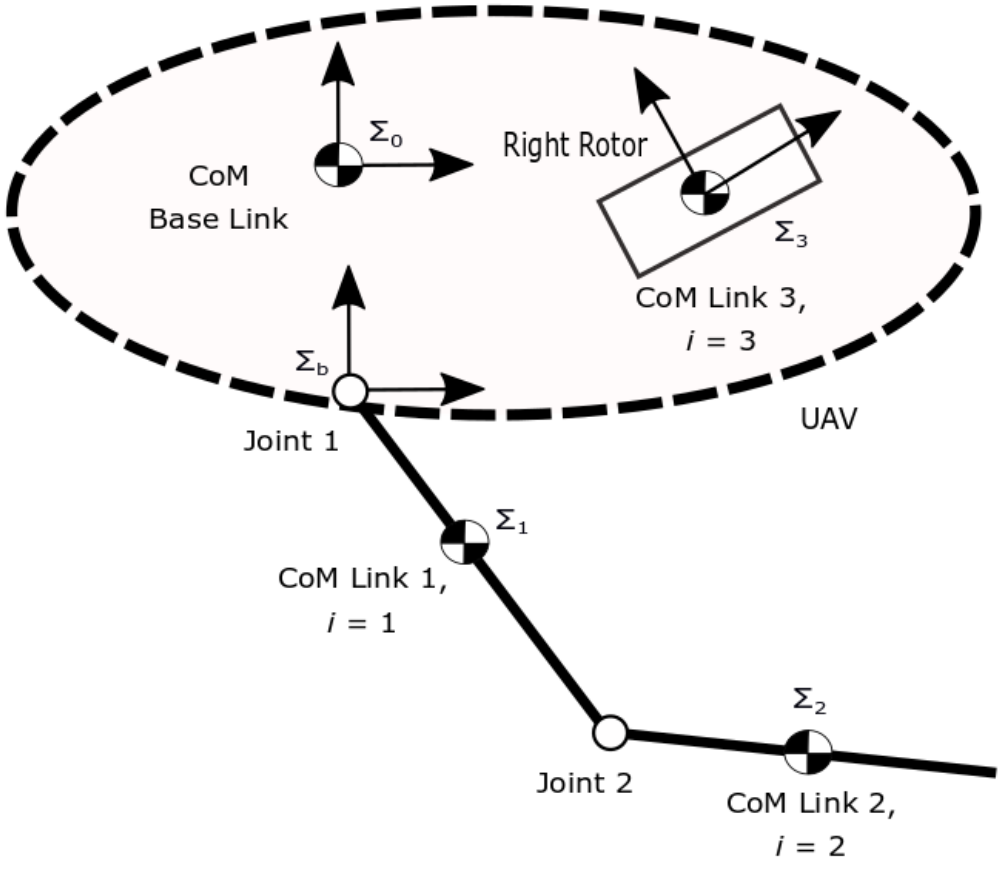


Figure 6.1: Links in the UAM.

The pose of the HM-UAM in its entirety can now be defined using a frame of reference placed at the CoM of the base link (the UAV) and aligning the axis of the corresponding reference frame of interest with the principle axis of the base link (i.e. Σ_0).

The current orientation of the base frame Σ_0 w.r.t the global (inertial) frame Σ_G is given by the rotation matrix R_0^G (defined in Chapter 5). As a result, the vehicle's current position can be given by the vector $X_0 = [x_0^G \quad y_0^G \quad z_0^G]^T$ which forms part of the state vector ζ. The velocity of the base link in the global frame is given by the vector $\dot{X}_0 = [\dot{x}_0\ \dot{y}_0\ \dot{z}_0]^T$ and its angular velocity about each axis in Σ_G coordinates is thus given by $\omega_G = \left[\ \omega_x\ \omega_y\ \omega_z\right]^T$.

The frame of reference of each link is centered on its CoM. The z-axis of such frames of reference is aligned with the link's rotational joint axis and the x-axis is assumed to be pointing towards the next joint axis as illustrated in Figure 6.2. The axis in which the joint rotates about is defined by the unit normal vector $u_i = \left[u_x\ u_y\ u_z\right]^T$ which is resolved in Σ_i coordinates. The position of a child link i w.r.t its parent link $\lambda(i)$ is described by the vectors, p_i, d_i and the current position of joint i (Figure 6.2). The vector p_i extends from the CoM of the parent link $\lambda(i)$ to joint i and the vector d_i extends from joint i to the CoM of Link i. Thus, the current position of joint i, which is defined as the angle between p_i and d_i, is given by the rotational angle θ_i.

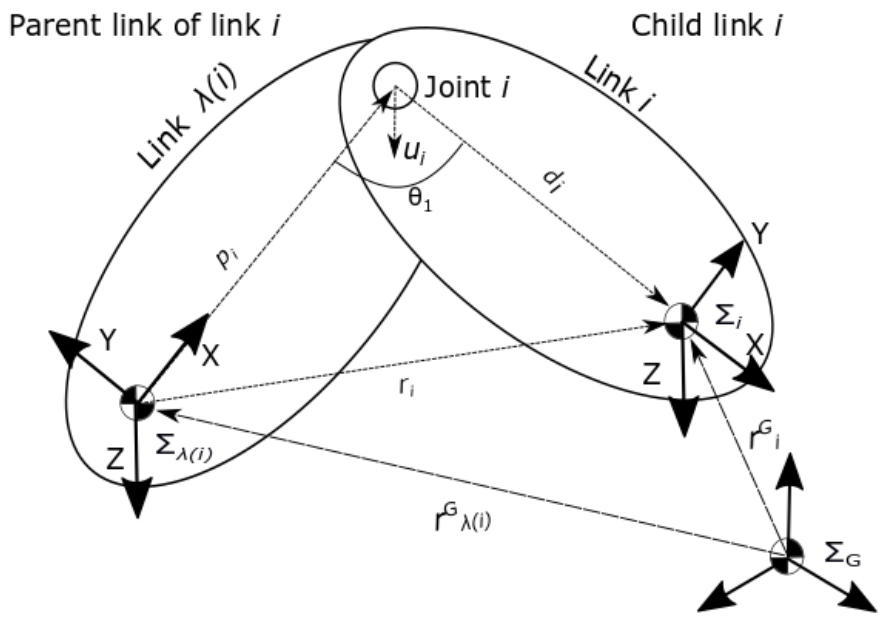


Figure 6.2: Diagram defining vectors for parent and child links.

The sum of external generalized forces applied to link i is transformed into a torque vector $\tau^{ext} = \left[\tau_x^{ext}\ \tau_y^{ext}\ \tau_z^{ext}\right]_i^T$ and a force vector $F^{ext} = \left[F_x^{ext}\ F_y^{ext}\ F_z^{ext}\right]_i^T$ acting on the CoM of the link. The joint actuator torque Q_i defined as a function of τ^{ext} and F^{ext}, $Q_i = f(\tau^{ext}, F^{ext})$ produced by joint i acts about its corresponding z-axis composing the joint axis u_i vector using the right-hand rule.

To obtain the complete dynamics the inertia of the links is required. In the multibody formulation used, the mass of a link i is treated as a point mass, and the corresponding inertial tensor I_i acting at Σ_i is defined in Σ_i coordinates as:

$$I_i = \begin{bmatrix} I_{x_i x_i} & I_{x_i y_i} & I_{x_i z_i} \\ I_{y_i x_i} & I_{y_i y_i} & I_{y_i z_i} \\ I_{z_i x_i} & I_{z_i y_i} & I_{z_i z_i} \end{bmatrix} \tag{6.2}$$

6.2 Kinematic Relationships

For the chosen multibody algorithm, the kinematic relations between links and their reference frames must be defined. The relations defined in Chapter 5 are used in this section as well as new relations being defined in what follows.

To rotate between the global frame and the UAV's frame of references, the rotation matrix R_0^G defined by Equation 5.1, is used. The rotation matrix that transforms from the parent frame $\Sigma_{\lambda(i)}$ to the non-base link frame Σ_i is given per Equation (6.3) and is based on the current angle of joint i, θ_i, and its axis of rotation u_i.

$$R_i^{\lambda(i)} = \begin{bmatrix} \cos(q_i) + u_{x,i}^2(1-\cos(q_i)) & u_{x,i}u_{y,i}(1-\cos(q_i)) + u_{z,i}\sin(q_i) & u_{x,i}u_{z,i}(1-\cos(q_i)) - u_{y,i}\sin(q_i) \\ u_{x,i}u_{y,i}(1-\cos(q_i)) - u_{z,i}\sin(q_i) & \cos(q_i) + u_{y,i}^2(1-\cos(q_i)) & u_{y,i}u_{z,i}(1-\cos(q_i)) + u_{x,i}\sin(q_i) \\ u_{x,i}u_{z,i}(1-\cos(q_i)) + u_{y,i}\sin(q_i) & u_{y,i}u_{z,i}(1-\cos(q_i)) - u_{x,i}\sin(q_i) & \cos(q_i) + u_{z,i}^2(1-\cos(q_i)) \end{bmatrix} \quad (6.3)$$

The rotation matrix defining the orientation of the global frame Σ_G w.r.t the frame of the non-base link i Σ_i is expressed as:

$$R_i^G = R_i^{\lambda(i)} R_{\lambda(i)}^G \quad (6.4)$$

The translation between reference frames can be defined by the vectors r_0 and r_i, where the vector r_0 extends from the global reference frame Σ_G to the base frame Σ_0, and the vector r_i (shown in Figure 6.2) extends from the origin of the frame of reference $\Sigma_{\lambda(i)}$ to the origin of the frame of reference Σ_i. These vectors are used to define the location of the base link, or each non-base link w.r.t their parent frame of reference, and can be found using the following two expressions:

$$r_0 = R_0^G X + d_0 \quad (6.5)$$

$$r_i = r_i^{\lambda(i)} = R_i^{\lambda(i)} p_i + d_i \tag{6.6}$$

The relative position of link i w.r.t the global frame Σ_G is given by the vector r_i^G which extends from Σ_G to Σ_i, and is calculated using the following expression:

$$r_i^G = R_i^G r_{\lambda(i)}^G + r_i \tag{6.7}$$

Now that the reference frames and their relations have been defined, the velocity of each link within the HM-UAM can be calculated. Starting with the base link, the linear velocity of the base Link 0 in base coordinates Σ_0 is defined as $v_0 = \begin{bmatrix} v_x & v_y & v_z \end{bmatrix}^T$ and expressed as:

$$v_0 = R_0^G \dot{X}_G \tag{6.8}$$

where $\dot{X}_G$ is the velocity of the HM-UAM defined in terms of Σ_G. The generalized velocity, $\hat{v}_0$, of the base link (position and rotations) can then be expressed as:

$$\hat{v}_0 = \begin{bmatrix} \omega_0 \\ v_0 \end{bmatrix} = \begin{bmatrix} \omega_0 \\ R_0^G \dot{X}_G \end{bmatrix} \tag{6.9}$$

The spatial transformation matrix used to transform generalized forces, accelerations and velocities from the parent link's frame $\Sigma_{\lambda(i)}$ to the frame of link i Σ_i is a 6x6 matrix. The generalized spatial transformation matrix, $\hat{T}_i^{\lambda(i)}$, from the parent frame $\lambda(i)$ to the current frame i can be expressed as:

$$\hat{T}_i^{\lambda(i)} = \begin{bmatrix} R_i^{\lambda(i)} & 0_{3\times3} \\ -r_i \times R_i^{\lambda(i)} & R_i^{\lambda(i)} \end{bmatrix} \tag{6.10}$$

The reverse transform, $\hat{T}_i^{\lambda(i)}$, from the joint i to the parent frame can then be expressed as:

$$\hat{T}_i^{j*} = \begin{bmatrix} R_j^{i^T} & R_j^{i^T} r_j \times \\ 0_{3\times3} & R_j^{i^T} \end{bmatrix} \tag{6.11}$$

where $r_i \times$ is the skew symmetric operator applied to r_i expressed as:

$$r_i \times = \begin{bmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{bmatrix} \tag{6.12}$$

The spatial joint axis operator $\hat{S}_i$ for joint i used to transform from the joint axis to a generalized vector acting at Σ_i is expressed as:

$$\hat{S}_i = \begin{bmatrix} u_i \\ u_i \times d_i \end{bmatrix} \tag{6.13}$$

An example for the use of $\hat{S}_i$ is transforming the joint velocity q_i to a generalized velocity as shown by Eqn (6.14). The generalized velocity of link i defined in Σ_i can be expressed as:

$$\hat{v}_i = \begin{bmatrix} \omega_i \\ v_i \end{bmatrix} = \hat{X}_i \, \hat{v}_{\lambda(i)} + \hat{S}_i \, \dot{q}_i \tag{6.14}$$

To calculate the spatial Coriolis acceleration $\hat{c}_i$ for joint i, first values for sub-calculations, B_i Eqn (6.15) and C_i Eqn (6.16) are needed.

$$B_i = \left(R_i \omega_{\lambda(i)}\right) \times \tag{6.15}$$

$$C_i = u_i \times d_i \tag{6.16}$$

where B_i and C_i represent sub-calculations for $\hat{c}$ which are used to create a more compact and efficient form of $\hat{c}$. Then the Coriolis acceleration of link i can then be computed using Eqn 6.17:

$$\hat{c}_i = \begin{bmatrix} B_i u_i \dot{q}_i \\ B_i B_i r_i + 2 B_i C_i \dot{q}_i + u_i \times C_i \dot{q}_i^2 \end{bmatrix} \tag{6.17}$$

The generalized acceleration of link i $\hat{a}_i$, can be computed as:

$$\hat{a}_i = \hat{T}_i^{\lambda(i)} \hat{a}_{\lambda(i)} + \hat{S}_i \ddot{q}_i + \hat{c}_i \tag{6.18}$$

With this the first pass of the RNEMD algorithm is completed and the algorithm can now proceed to its second pass.

6.3 Generalized Forces and Inertias (Second Pass).

The second pass in the algorithm works inwards recursively from the last link n, and moves inwards in decreasing link number finishing with the base link. In the second pass, the generalized forces, and inertial effects are calculated for each of the links comprising the HM-UAM. The variables that need to be precomputed for this pass include the spatial inertial matrix $\hat{I}_i$ for each link i, and the (6x1) spatial bias force vector $\hat{Z}_i$, also for each link, which are defined as follows:

$$\hat{I}_i = \begin{bmatrix} I_i & 0_{3\times3} \\ 0_{3\times3} & m_i E_{3\times3} \end{bmatrix} \tag{6.19}$$

$$\hat{Z}_i = \begin{bmatrix} \omega_i \times I_i \omega_i \\ 0_{3\times1} \end{bmatrix} \tag{6.20}$$

The recursive algorithm for the second pass starts by finding the spatial articulated inertia for each of the links. $\hat{I}_i^A$ which defines the combined inertia of link i and its child links $\mu(i)$ acting at the CoM of link i:

$$\hat{I}_i^A = \hat{I}_i + \sum_{j\in\mu(i)} \hat{T}_i^{j*} \left(\hat{I}_j^A - P_j U_j^T\right) \hat{T}_j^i \tag{6.21}$$

where U_j(Eqn 6.23) and P_j (Eqn 6.25) are terms that reoccur throughout the recursive algorithm.

$$U_i = \hat{I}_j^A \hat{S}_j \tag{6.22}$$

$$P_j = U_j D_j \tag{6.23}$$

The next step in the algorithm is to compute the spatial articulated bias force $\hat{Z}_i^A$ which is the generalized forced acting on link i due to the motion of the HM-UAM, which is accomplished via Eqn. 6.24.

$$\hat{Z}_i^A = \hat{Z}_i - \hat{F}_i^{ext} + \sum_{j \in \mu(i)} T_i^{j*}\left(P_j\left(Q_j - K_j\right) + H_j\right) \tag{6.24}$$

Included in the external forces applied to each link is the force due to gravity for that link. Furthermore, to calculate both, $\hat{I}_i^A$, and $\hat{Z}_i^A$, the shared sub-calculations D_i, H_i, K_i are first needed and are computed as follows:

$$D_i = \left(\hat{S}_i^T U_i\right) \tag{6.25}$$

$$H_j = \hat{I}_j^A \, \hat{c}_j + \hat{Z}_j^A \tag{6.26}$$

$$K_j = \hat{S}_j^T H_j \tag{6.27}$$

The total force generalized force acting on link i given by Eqn (6.1) can now be determined provided the generalized acceleration of link i, $\hat{a}_i$, can be computed. $\hat{a}_i$ can be obtained via Equation 6.18. The force acting on link i as defined by Eqn 6.1 can be transformed onto the joint axis u_i using the spatial joint axis operator $\hat{S}_i$. The resulting joint torque due to the spatial articulated inertia of the link and the spatial articulated bias force is then found via following equation 6.28:

$$Q_i = \hat{S}_i^T \hat{F}_i^I \tag{6.28}$$

Expanding this equation, the following is obtained:

$$Q_i = \hat{S}_i^T \hat{I}_i^A \hat{S}_i \ddot{q}_i + \hat{S}_i^T \hat{I}_i^A \hat{T}_{\lambda(i)}^i \hat{a}_{\lambda(i)} + \hat{S}_i^T \left(\hat{I}_i^A \hat{c}_i + \hat{Z}_i^A\right) \tag{6.29}$$

The second pass finishes with the base link (link 0).

The complete algorithm used to calculate $Q_1, Q_2, \hat{F}_0^I$ via Equations 6.1, 6.21, 6.22, and 6.29 is schematically shown in Figure 6.3.

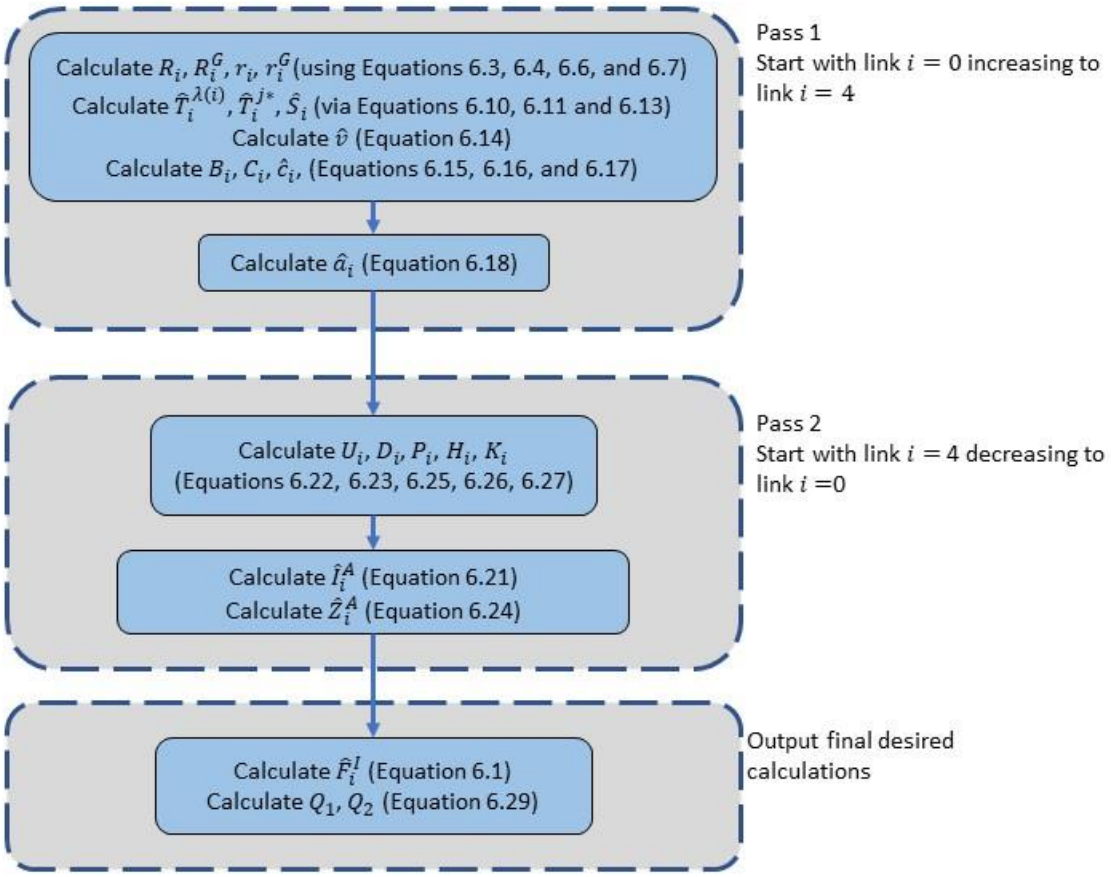


Figure 6.3: Multibody Dynamics Algorithm.

6.4 Conclusions

In this Chapter, a formulation for the dynamic model of the HM-UAM is developed using the recursive Newton-Euler multibody dynamics proposed in [49]. Using the formulation, the generalized force acting on each link is found and transformed to the link's joint axis which enables to find the corresponding resultant torque. The next chapter, Chapter 7, will use the mathematical formulations developed in this Chapter to develop the control of the HM-UAM.

Chapter 7 - Control Algorithm

7.1 Control Algorithm Overview

This chapter describes the proposed control algorithm for the UAM. The control architecture, depicted in Figure 7.1, is devised of several parts: the inverse kinematic model, the UAM controller uses an adaptive sliding mode controller, and the inverse dynamics for the UAV and arm. The goal of the control algorithm is to take a given 3D reference trajectory for the end-effector as an input and produce the control outputs (i.e., main propellers tilting angles, arm joint angles, ect) for the HM-UAM. The inverse kinematics model takes the reference trajectory for the end-effector comprising the HM-UAM as an input and outputs the modified desired state vector q_{des} and its derivatives $\dot{q}_{des}$ and $\ddot{q}_{des}$ for the HM-UAM. The state vector is then converted into a modified state vector q^* inside the controller using the current x_0 and $\dot{x}_0$ values. From the modified state vector, the HM-UAM controller outputs a desired reference acceleration $\ddot{q}_r$ to be used by the dynamic model. The inputs for the dynamic model (Chapter 5) are the current state of the HM-UAM q and $\dot{q}$ and the reference acceleration $\ddot{q}_r$ and outputs the required joint torques and forces that will take the HM-UAM to its desired state, q^*. The HM-UAM controller's final output is the desired joint torques for the robotic arm (i.e., τ_1, τ_2), and the desired forces and torques that need to be generated by the HM-UAV propeller system for the HM-UAV (i.e. $F_{y_0}, F_{z_0}, \tau_{x_0}, \tau_{y_0}, \tau_{z_0}$). The inverse dynamics model for the HM-UAV is then used to convert these values into the corresponding joint (propeller tilting) angles, q_3 and q_4, and the needed thrust values, F_3, F_4 and F_5, for the propellers comprising the HM-UAV.

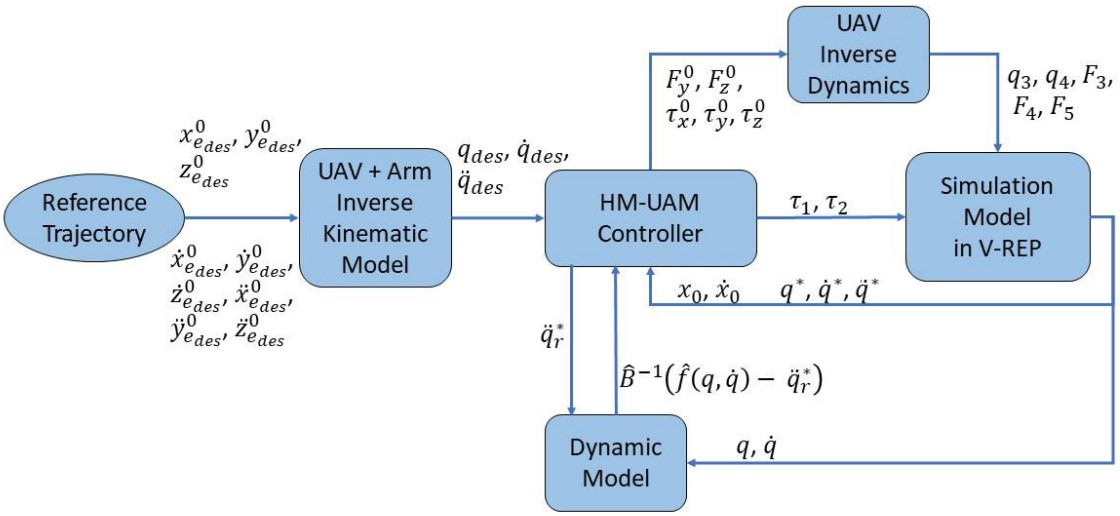


Figure 7.1: Control Algorithm Flow Chart.

7.2 Inverse Kinematics

The controller for the UAM uses the desired arm joint positions and desired UAV position and orientation as inputs These are generated from the inverse kinematics based on the desired reference trajectory that the arm is required to follow/track. However, the desired trajectory for the end-effector is typically provided in global coordinates and first need to be converted in UAV coordinates Σ_0. The inverse kinematics formulation/representation (shown in Fig 7.2) is used to convert the desired end-effector trajectory in Σ_0 into the joint space.

The UAM is under-constrained for following trajectories in the plane of the attached robot arm. If the desired trajectory for the end-effector is within the plane in which the robotic arm operates in (i.e., the vertical longitudinal plane of the HM-UAV shown Figure 3.1), the HM-UAM has more DoF (i.e., four DOF) than it has constraints (i.e., two constraint) and acts like a redundant manipulator. As a result, if the HM-UAV is considered part of the arm's support base

(shoulder) it acts like two prismatic joints that are orthogonal to each other in the arm's inverse kinematics with on of the prismatic joints parallel to the arm's original rotation shoulder joint (i.e., θ_1 in Figure 4.6). To ease the inverse kinematic solution due to the high number of DOF of the HM-UAV + arm, it is suggested to constrain it. For this the system is constrained when the inverse kinematics is complex to solve for part of the desired trajectory that is being currently followed. This occurs when the trajectory is not near the limits of the arm's workspace. In such cases the desired motion of the UAV in the plane in which the robotic arm operates is minimal. Otherwise, the UAV needs to move (translate and/or rotate) to bring the next part of the trajectory to be followed within the arm's workspace.

The above described procedure is done because if the UAV moves, it creates large reaction torques and forces in the robotic arm which increases the arm's needed control effort. Furthermore, if the UAV moves it changes the reference trajectory that the arm is required to follow as well. Under these conditions the cooperation between the UAV and the arm is reduced making it simpler to control the overall system while still making it possible to use the system in real applications (assuming the control of the HM-UAV is accurate).

In general terms, the desired position of the end-effector identified by $p^0_{e_{des}}$ is a point in 3D space. However, as the robot arm is constrained to move in the longitudinal plane of the UAV, identified by the Z_0-Y_0 plane of the UAV's frame of reference (Figure 3.1). As a result, any 3D point that is outside of such plane will require the UAV to move in such a way as such point is positioned within the arm's workspace. To reach any 3D points that have an x component different from x = 0, the UAV must change its orientation by yawing until the desired point is in the workspace plane of the arm. The amount that the UAV needs to yaw denoted as $\theta_{yaw_{des}}$ is determined by the angle between the vector $p^b_{e_{des}}$ and the unit vector Y_b shown in

figure 1. At this time, it is assumed that the 3D point of interest can be reached by the arm after the UAV has yawed by $\theta_{yaw_{des}}$ and the 3D point is within the 2D workspace of the arm. If this is not the case the UAV will have to perform additional translational motions (e.g., descend) until the point of interest (now within the longitudinal plane of the UAV) lies within the arm's workspace.

If and when the desired point within the trajectory that the end-effector is commanded to follow is not 100% inside the arm's workspace (in the plane in which the arm operates) the arm could still make an attempt to follow such trajectory at the same time the UAV maneuvers (e.g., yaws). To do this the desired trajectory is projected into the plane of the arm (i.e., UAV longitudinal plane) which will define the curve (e.g., point) that the arm should track.

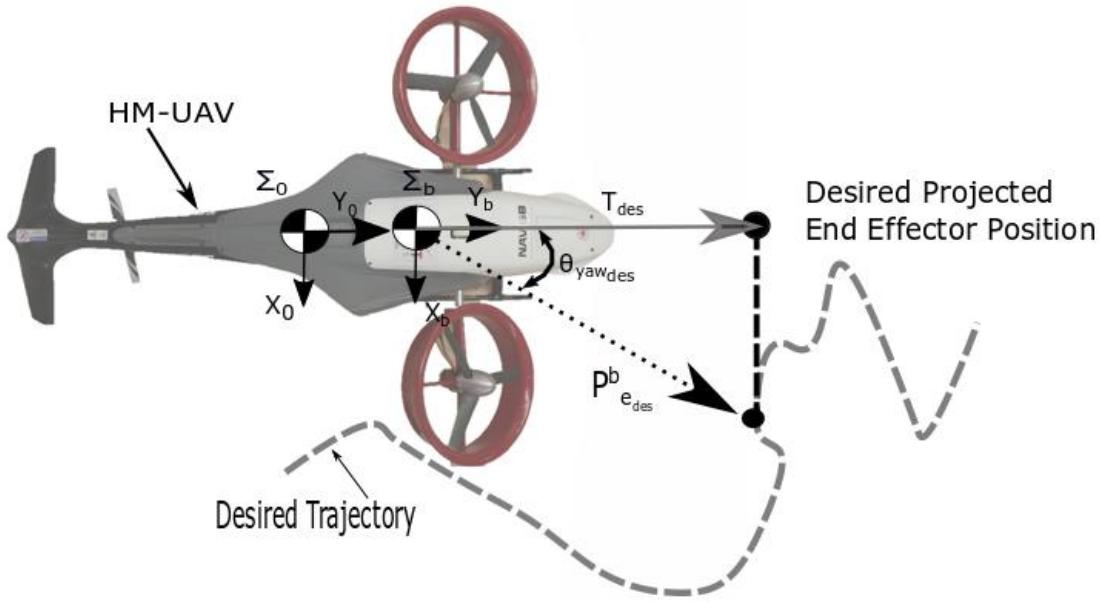


Figure 7.2: Top down view of the HM-UAM following a trajectory in the X_0-Y_0 plane. The projection of the desired trajectory into the plane of the arm is shown.

The vector between the base of the arm and the projected desired position of the end-effector defined in the arm's base coordinates Σ_b will be denoted as $\vec{T}^b_{des}$ In Figure 7.2 the

projection of $\vec{T}_{des}^{b}$ onto the y_b axis is denoted as T_{des}. As a result, the vector $\vec{T}_{des}^{b}$ is always in the $Y^b - Z^b$ plane (i.e., the longitudinal/plane of the HM-UAV) expressed as:

$$\vec{T}_{des}^{b} = \begin{bmatrix} y_{e_{des}}^{b} \\ z_{e_{des}}^{b} \end{bmatrix} \tag{7.1}$$

The angle between the x_b unit vector and the vector $\vec{T}_{des}^{b}$ is denoted as θ_T and written as:

$$\theta_T = 2\pi - \text{atan2}^{-1}\left(\frac{y_{e_{des}}^{b}}{z_{e_{des}}^{b}}\right) \tag{7.2}$$

where $x_{e_{des}}^{b}$, $y_{e_{des}}^{b}$, $z_{e_{des}}^{b}$ are the cartesian position coordinates of the end-effector in the manipulator's base's coordinates Σ_b. The magnitude of $\vec{T}_{des}^{b}$ with respect to the UAV's frame of reference is thus written as:

$$\left|T_{des}^{b}\right| = \sqrt{\left(y_{e_{des}}^{b}\right)^2 + \left(z_{e_{des}}^{b}\right)^2} \tag{7.3}$$

The maximum total length of the manipulator, $L_{arm\,max}$, can be expressed as:

$$L_{arm\,max} = L_1 + L_2 \tag{7.4}$$

where L_1, and L_2 are the lengths of Links 1 and 2. To perform the inverse kinematics for the HM-UAM, the controller must consider the scenario where the desired trajectory that the arm's end-effector is commanded to track is outside of the workspace of the arm. To accomplish this the proposed algorithm uses a piecewise function that is dependant on the magnitude of the vector $\left|T_{des}^{b}\right|$ which in turn is used to check the distance that the desired trajectory point is from the base of the arm. For example, if the desired point is outside of the workspace of the arm, that is to say $\left|T_{des}^{b}\right| \geq L_{arm\,max}$, then the UAV will need to move along the vector T_{des}^{b} until the trajectory (point) is within the workspace of the arm. Other scenarios where movement of the UAV is required include when the arm is near a singularity the UAV moves to increase the manipulability of the arm. When movement of the UAV is required and as part of the inverse

dynamics, the vector T_{des}^{b} is modified and the new desired position of the end-effector that is within the workspace of the arm is given by the vector $T_{des_a}^{b}$. To prevent the arm from reaching its singularity point a modified trajectory for the manipulator $T_{des_a}^{b}$ is used.

The piecewise functions for the desired position of the end-effector $T_{des_a}^{b}$ in manipulator base coordinates Σ_b, and the desired position of the UAV P_{UAV} in UAV coordinates Σ_0 are expressed as follows:

$$T_{des_a}^{b} = \begin{cases} T_{des}^{b}, & if \quad |T_{des}^{b}| < L_{arm\,max} \\ T_{des_a}^{b}, & if \quad |T^{b}{}_{des}| \geq L_{arm\,max} \end{cases} \tag{7.5}$$

$$P_{UAV_a}^{0} = \begin{cases} p_G^0 + 0.1\, L_{arm\,max} & if \quad T_{des}^{b} < L_{arm\,max}^{b} \\ p_G^0 + \left(T_{des}^{b} - L_{arm\,max} \right) & if \quad T_{des}^{b} \geq L_{arm\,max}^{b} \end{cases} \tag{7.6}$$

To solve for the desired arm joint positions θ_1, θ_2, both $T_{des_a}^{b}$ and θ_T are needed. The joint positions, θ_1, and θ_2, are found using the cosine law for the triangle made up of Link 1, Link 2 and $T_{des_a}^{b}$ as illustrated in Figure 7.3.

$$\theta_{1a} = \cos^{-1} \frac{-T_{des}^{b}{}^{2} - (L_1)^2 + (L_2)^2}{-2 * T_{des}^{b} * L_1} \tag{7.7}$$

$$\theta_1 = \theta_T + \theta_{1a} \tag{7.8}$$

$$\theta_{2a} = cos^{-1} \frac{-(L_1)^2 - (L_2)^2 + T_{des}^{b}{}^{2}}{-2L_1L_2} \tag{7.9}$$

$$\theta_2 = \pi - \theta_{2a} \tag{7.10}$$

The next step of the inverse kinematics solution is finding the robot arm joint velocities $\dot{\theta}_1$ and $\dot{\theta}_2$. To do this first the manipulator's velocity Jacobian is obtained and is written as:

$$J_v = \begin{bmatrix} -L_1 s_1 - L_2 s_{12} & -L_2 s_{12} \\ L_1 c_1 + L_2 c_{12} & L_2 c_{12} \end{bmatrix} \tag{7.11}$$

where $s_1 = \sin(\theta_1)$, $s_{12} = \sin(\theta_1 + \theta_2)$, $c_1 = \cos(\theta_1)$ and $c_{12} = \cos(\theta_1 + \theta_2)$. In this work, the inverse of the manipulator velocity Jacobian is calculated numerically using Gaussian elimination and then used to find the joint velocities for the arm which are found using Eqn (7.12).

$$\dot{q}_{des} = J_v^{-1}\left(\dot{X}^b{}_{des} - V_{UAV}^b\right) \tag{7.12}$$

where $\dot{X}_{des}^b$ is the desired end-effector velocity, V_{UAV}^b is the velocity of the UAV, and $\dot{q}_{des}$ is the desired arm's joint velocity vector. The desired end-effector acceleration $\ddot{X}_{des}$ can then be expressed as:

$$\dot{X}_{des}^b = \frac{d}{dt}\dot{X}_{des}^b = \dot{J}_v\, \dot{q}_{des} + J_v \ddot{q}_{des} \tag{7.13}$$

$$\ddot{X}_{des}^b = \dot{J}_v\, \dot{q}_{des} + J_v \ddot{q}_{des} \tag{7.14}$$

Equation 7.13 can be solved for $\ddot{q}_{des}$ and the resulting equation is written as:

$$\ddot{q}_{des} = J_v^{-1}\left(\ddot{X}_{des}^b - \dot{J}_v \dot{q}_{des} - \dot{V}_{UAV}^b\right) \tag{7.15}$$

where $\dot{V}_{UAV}^b$ is the acceleration of the UAV in Σ_b. The desired rotation rate for the UAV, $\omega_{y_{des}}^0$, is based on $\dot{x}_{e_{des}}^b$ and written as:

$$\omega_{y_{des}}^0 = \frac{\dot{x}_{e_{des}}^b}{y_{e_{des}}^b} \tag{7.16}$$

At this point the desired trajectory that the arm is commanded to follow/track (in its own workspace) and the corresponding required position and motions of the HM-UAV and arm have been determined. Thus, all information required to control the HM-UAV and arm is available.

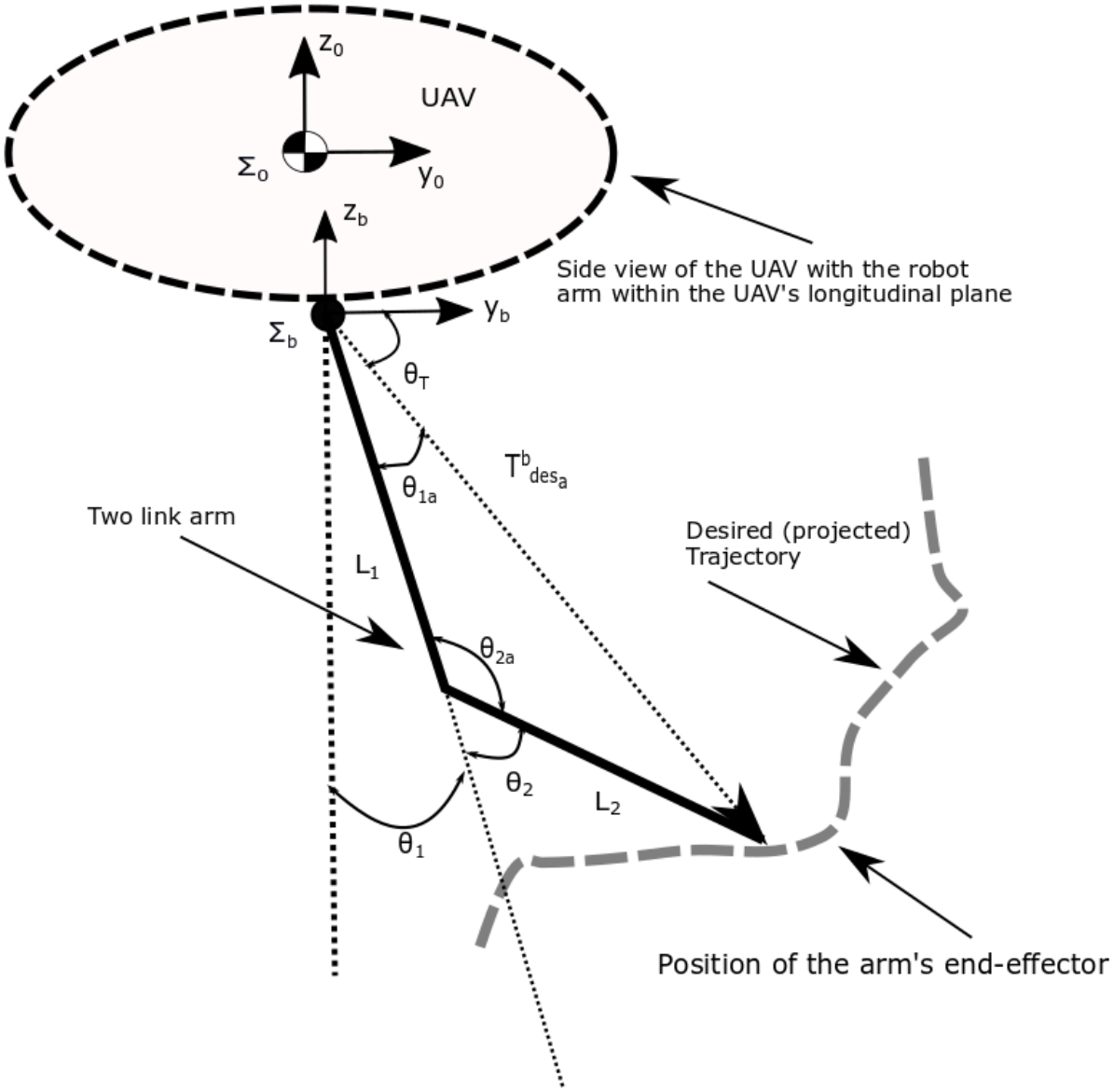


Figure 7.3: Arm inverse kinematics.

7.3 HM-UAM controller

From the mathematical model of the UAV and the arm (Section 6.3) it is identified that the HM-UAM is a multiple input multiple output (MIMO) second order nonlinear control affine system. As a result, the proposed control model is given by the following expression:

$$\ddot{q} = f(q, \dot{q}, t) + B(q, t)\, u(q, \dot{q}, t) \tag{7.17}$$

where $f(q,\dot{q},t)$ is the vector function of the state vector that represents the non-linear dynamics of the system, $B(q,t)$ is the control gain matrix, $u(q,\dot{q},t)$ is the control output, $q = [q_1,q_2,q_3,q_4,q_5,q_6,q_7,q_8]^T$ is the vector of generalized coordinates, $\dot{q} = [\dot{q}_1,\dot{q}_2,\dot{q}_3,\dot{q}_4,\dot{q}_5,\dot{q}_6,\dot{q}_7,\dot{q}_8]^T$, and $\ddot{q} = [\ddot{q}_1,\ddot{q}_2,\ddot{q}_3,\ddot{q}_4,\ddot{q}_5,\ddot{q}_6,\ddot{q}_7,\ddot{q}_8]^T$ are the velocity and acceleration of the generalized coordinates respectively. Table 6.1 provides the specific physical description for each of the generalized coordinates, q_i.

Table 7.1: HM-UAM State Variables.

Generalized Coordinate Number	Description
$\boldsymbol{q_1}$	X position of UAV in Σ_0 (x_0)
$\boldsymbol{q_2}$	Y position of UAV in Σ_0 (y_0)
$\boldsymbol{q_3}$	Z position of UAV in Σ_0 (z_0)
$\boldsymbol{q_4}$	Pitch (β)
$\boldsymbol{q_5}$	Yaw (γ)
$\boldsymbol{q_6}$	Roll (α)
$\boldsymbol{q_7}$	Position of Joint 1 of Arm (θ_1)
$\boldsymbol{q_8}$	Position of Joint 2 of Arm (θ_2)

The chosen controller for the HM-UAM is an adaptive sliding mode controller with an uncertainty estimator developed in [51] (there are some doubts whether this controller is truly an adaptive sliding mode controller but this controller will be referred to as a adaptive sliding mode controller for the purpose of simplicity). The schematic representation of such controller is shown in Figure 6.2. The advantages of using a sliding mode control (SMC) vs other types of controllers include its ability to deal with modelling uncertainties, small tracking error, and fast response [51]. Disadvantages of typical adaptive sliding mode controllers include slow adaption time, and slower controller dynamics due to the adaption and chattering aspects known to exist in such controllers. To overcome such disadvantages in the proposed controller, an uncertainty estimator vector, $\tilde{F}_{est}$, is used to adapt the controller to unmodeled dynamics. The uncertainty

estimator is used for adaption instead of a parameter adaptive controller. Due to the increased speed of adaption and lower computational time. Lower adaption speed can result in stability and robustness issues. Unlike most SMC which eliminates the chattering that is typical in SMCs that use a discontinuous control law the chosen control law is continuous. The selection to use such controller architecture is due to the fact that there is an available Lyapunov function that has show effectiveness [51]. In what follows such Lyapunov function is used. The Lyapunov stability proof for the proposed control law (Eqn 7.20) for a system described by Eqn (6.29) is given in [51]. As a result, the chosen control law is given as:

$$u(t) = \hat{B}^{-1}\left(\ddot{q}_r - \hat{f} - Ks - \tilde{F}_{est}\right) \tag{7.18}$$

where $\hat{f}$ is the known part of the inverse dynamics, $\tilde{F}_{est}$ is the estimated perturbation vector, K is a diagonal positive definite constant matrix, $\ddot{q}_r$ is the reference acceleration, s is the sliding surface, and $\hat{B}^{-1}$ is the known part of the control gain matrix. The controller in Figure 7.4 represents the internal structure of the controller box shown in Figure 7.1.

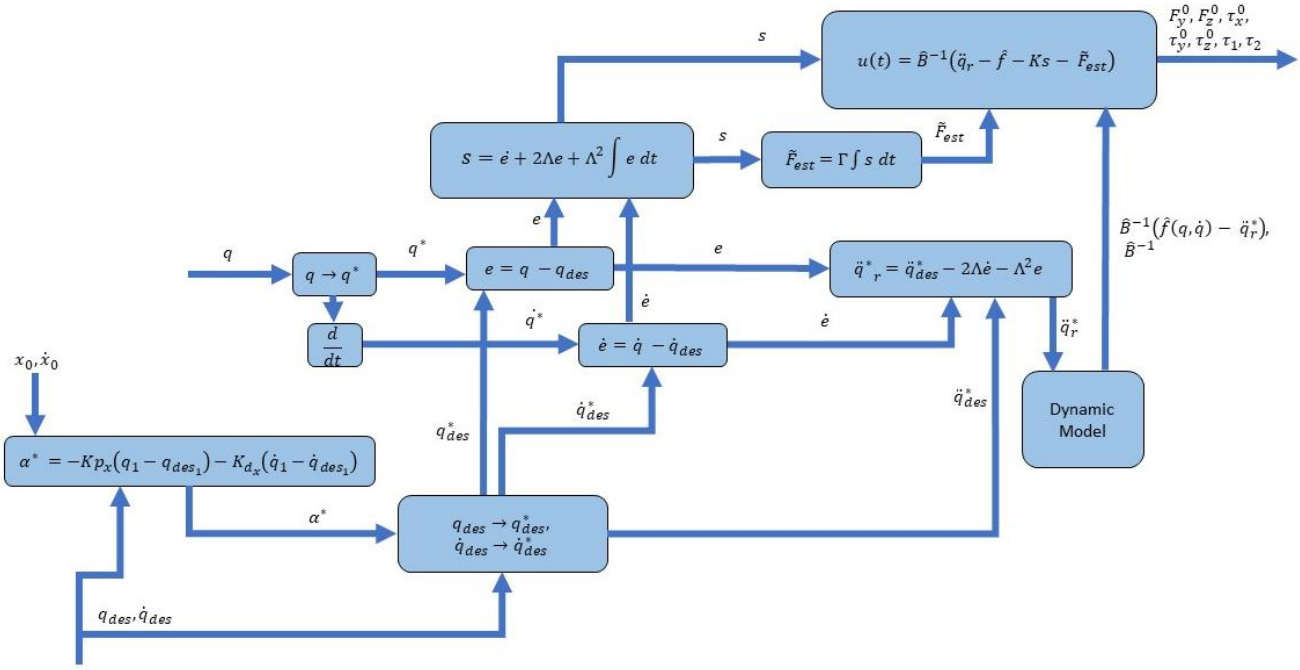


Figure 7.4: UAV and Arm Controller.

As described in Section 4.1, the acceleration in the x_0 direction of the HM-UAV is uncontrollable (coupled with its roll motion), therefore to overcome the lateral position errors (occurring in the x_0 direction of the HM-UAV) the UAV controller will be required to change its desired roll based on the X error. When doing this, the HM-UAV will tilt the workspace plane in which the robot arm operates potentially introducing other errors in the position of the end-effector. The desired roll motion is thus described by Eqn 7.19.

$$Roll_{des_x} = -K_{p_x}(q_1 - q_{des_1}) - K_{d_x}(\dot{q}_1 - \dot{q}_{des_1}) \tag{7.19}$$

where K_{p_x} and K_{d_x} are positive control gains, q_{des_1} is the desired x^0 position of the UAV, and $\dot{q}_{des_1}$ is the desired x^0 velocity for the UAV. Since $\ddot{q}_1$ (lateral acceleration), is not controllable the reduced control input q^* Eqn 7.20 will be used.

$$q^* = [q_2 \quad q_3 \quad q_4 \quad q_5 \quad q_6 \quad q_7 \quad q_8]^T \tag{7.20}$$

The controller (Figure 7.4) receives two general inputs (i.e., $[x_0, \dot{x}_0, q^*, \dot{q}^*, \ddot{q}^*]$ and $\hat{B}^{-1}\left(\hat{f}(q,\dot{q}) - \ddot{q}_r^*\right)$) and generates three general outputs (i.e., $[\tau_1, \tau_2]$, $\ddot{q}_r^*$ and $[F_y^0, F_z^0, \tau_x^0, \tau_y^0, \tau_z^0]$) as illustrated in Figure 7.1 and Figure 7.4. The chosen sliding surface is a PID type controller that is used to increase the closed loop performance and improve the performance while in the learning process. The sliding surface used in this work is defined as:

$$s = \dot{e} + 2\Lambda e + \Lambda^2 \int e \, dt \tag{7.21}$$

where Λ is a positive definite diagonal gain matrix, e and $\dot{e}$ is the generalized coordinate error, and its derivative. For the proposed aerial manipulator, the generalized coordinates vector error are written as:

$$e = q^* - q_{des}^* \tag{7.22}$$

$$\dot{e} = \dot{q}^* - \dot{q}_{des}^* \tag{7.23}$$

The derivative of s is written as:

$$\dot{s} = \ddot{q}^* - (\ddot{q}_{des}^* - 2\Lambda\dot{e} - \Lambda^2 e) \tag{7.24}$$

The reference acceleration $\ddot{q}_r$ uses the second half of the derivative of the sliding surface and is defined as:

$$\ddot{q}_r^* = (\ddot{q}_{des}^* - 2\Lambda\dot{e} - \Lambda^2 e) \tag{7.25}$$

The control output due to the inverse dynamics $\hat{B}^{-1}\hat{f}(q,\dot{q})$ for the generalized coordinates representing the HM-UAV (q_i for $i = 2, \ldots, 5$) representing F_y, F_z, τ_x, τ_y, and τ_z can be represented by the generalized force, $\hat{F}_0^I$, acting on the UAV. For each link in the robotic arm the inverse dynamics are represented by Eqn (6.29). As a result, the combined term for the control output for the combined inverse dynamics and the reference acceleration for the UAS can be written as follows:

$$\hat{B}^{-1}\left(\hat{f}(q,\dot{q}) - \ddot{q}_r^*\right) = \begin{bmatrix} \hat{F}_0^I(4) \\ \hat{F}_0^I(5) \\ \hat{F}_0^I(6) \\ \hat{F}_0^I(2) \\ \hat{F}_0^I(3) \\ \hat{S}_0^T \hat{I}_1^A \hat{S}_1 \ddot{q}_r^*(1) + \hat{S}_1^T \hat{I}_1^A \hat{T}_{\lambda(1)}^i \hat{a}_{\lambda(1)} + \hat{S}_1^T\left(\hat{I}_1^A \hat{c}_1 + \hat{Z}_1^A\right)) \\ \hat{S}_1^T \hat{I}_2^A \hat{S}_2 \ddot{q}_r^*(2) + \hat{S}_2^T \hat{I}_2^A \hat{T}_{\lambda(2)}^i \hat{a}_{\lambda(2)} + \hat{S}_2^T\left(\hat{I}_2^A \hat{c}_2 + \hat{Z}_2^A\right)) \end{bmatrix} \tag{7.26}$$

The error in modeling the inverse dynamics for the HM-UAM $\tilde{f}$ is expressed as:

$$\tilde{f} = f - \hat{f} \tag{7.27}$$

where f is the real inverse dynamics for the HM-UAM, and $\hat{f}$ is the known (modeled) part of the inverse dynamics. Similarly, the modeling error for the control gain matrix, $\tilde{B}$, is expressed as:

$$\tilde{B} = B - \hat{B} \tag{7.28}$$

where B is the control gain matrix, and $\hat{B}$ is the known part of the control gain matrix.

The dynamics equation for the HM-UAS in its entirety can be expressed in the following form:

$$M(q^*)\ddot{q}^* + C(q^*,\dot{q}^*)\dot{q}^* + G(q^*) = \tau + \tau_{ext} \tag{7.29}$$

where $M(q^*) \in \mathbb{R}^{7\times7}$ is the inertia matrix which is positive definite and symmetric. $C(q^*,\dot{q}^*)$ is the Coriolis matrix and $C(q^*,\dot{q}^*)\dot{q}^*$ represent the Coriolis and Centrifugal terms, $G(q^*)$ is the gravity terms, τ_{ext} is the external forces/torques, and τ is the control input. Comparing (Eqn 7.29) to (Eqn 7.17), the non-linear dynamics term $f(q^*,\dot{q}^*,t)$ is a function of $C(q^*,\dot{q}^*)\dot{q}^*$, $G(q^*)$, and τ_{ext}, and the control input τ is the same as $u(q^*,\dot{q}^*,t)$. After this analysis it is then possible to see that the inverse of the control gain matrix B^{-1} is equal to the inertia matrix $M(q)$.

$$M(q^*)\,\ddot{q}^* = B(q^*)^{-1}\,\ddot{q}^* \tag{7.30}$$

Therefore the $\hat{B}^{-1}$ matrix Eqn (7.29) is based on the spatial articulated inertia of the UAV and Links 1 and 2 and is expressed as follows:

$$\hat{\mathrm{B}} = \begin{bmatrix} \hat{I}_0^A(2,2) & \hat{I}_0^A(2,3) & \hat{I}_0^A(2,4) & \hat{I}_0^A(2,5) & \hat{I}_0^A(2,6) & 0 & 0 \\ \hat{I}_0^A(3,2) & \hat{I}_0^A(3,3) & \hat{I}_0^A(3,4) & \hat{I}_0^A(3,5) & \hat{I}_0^A(3,6) & 0 & 0 \\ \hat{I}_0^A(3,2) & \hat{I}_0^A(4,3) & \hat{I}_0^A(4,4) & \hat{I}_0^A(4,5) & \hat{I}_0^A(4,6) & 0 & 0 \\ \hat{I}_0^A(4,2) & \hat{I}_0^A(5,3) & \hat{I}_0^A(5,4) & \hat{I}_0^A(5,5) & \hat{I}_0^A(5,6) & 0 & 0 \\ \hat{I}_0^A(5,2) & \hat{I}_0^A(6,3) & \hat{I}_0^A(6,4) & \hat{I}_0^A(6,5) & \hat{I}_0^A(6,6) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \hat{S}_1^T \hat{I}_1^A \hat{S}_1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \hat{S}_2^T \hat{I}_2^A \hat{S}_2 \end{bmatrix} \tag{7.31}$$

The adaptive part of the controller attempts to adapt to the modeling error in the inverse dynamics and the control gain matrix. In this controller, the adaption is performed using a single perturbation vector. The update law for the estimated perturbation $\dot{\tilde{F}}_{est}$, is used to update the estimated perturbation error $\tilde{F}_{est}$ at each controller time step and is written as:

$$\dot{\tilde{F}}_{est} = \Gamma \mathrm{s} \tag{7.32}$$

where Γ shown in Equation 7.31 is a diagonal positive constant matrix representing the learning rate.

$$\Gamma = \begin{bmatrix} 0.001 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.0002 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0001 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2.0 \end{bmatrix} \tag{7.33}$$

The error in estimating the true modeling error E is defined as:

$$E = \tilde{F}_{est} - \tilde{F} \tag{7.34}$$

where $\tilde{F}_{est}$ and $\tilde{F}$ are the estimated and the unknown perturbation errors respectively. The output of the control law given in Eqn (7.18), is in the following form:

$$u(t) = \begin{bmatrix} F_y^0 \\ F_z^0 \\ \tau_x^0 \\ \tau_y^0 \\ \tau_z^0 \\ \tau_1 \\ \tau_2 \end{bmatrix} = \hat{B}^{-1}\left(\ddot{q}_r - \hat{f} - Ks - \tilde{F}_{est}\right) \tag{7.35}$$

From Eqn 7.33, the HM-UAM can directly use the torque outputs τ_1 and τ_2 to guide the robot arm joints but will need to convert the remaining forces (F_y^0, and F_z^0) and torques (τ_x^0, τ_y^0, and τ_z^0) into the desired force vectors for the propellers moving the UAV. These desired forces and torques are defined in Σ_0 and act at the CoM of the UAV. The controller depends on the two main tilt propellers and the tail propeller to create the desired forces from the HM-UAV controller. Using the inverse UAV dynamics found in Section 7.3 (Equations 7.43, 7.44, 7.45, 7.46, 7.47, 7.48, 7.49, 7.50, 7.51, and 7.52), the desired forces and torques from the controller are converted into the tilting joint angles for the two main propellers and the magnitude of thrust required from each of the propellers.

7.4 Inverse Dynamics

The HM-UAM controller (Equation 7.33) gives the values for $\overrightarrow{F_0}$ and $\vec{\tau}_0$ for the UAV, and τ_1, and τ_2 for the robot arm as an output. The output forces and torques ($\overrightarrow{F_0}$ and $\vec{\tau}_0$) from the HM-UAM controller are expressed w.r.t the HM-UAV frame of reference Σ_0. The forces produced by each of the propellers ($\overrightarrow{F_3^0}$, $\overrightarrow{F_4^0}$, $\overrightarrow{F_5^0}$) are expressed as forces vectors w.r.t the Σ_0 HM-UAV frame of reference. Equation 7.34 shows the conversion of the thrust force vectors for the propellers ($\overrightarrow{F_3^0}$, $\overrightarrow{F_4^0}$, $\overrightarrow{F_5^0}$), to the resultant generalized force acting at the CoM of the HM-UAV.

$$\begin{bmatrix} \overrightarrow{F_0} \\ \vec{\tau}_0 \end{bmatrix} = \begin{bmatrix} R_3^{0T} & R_4^{0T} & R_5^{0T} \\ (\vec{r}_3 \times)\, R_3^0 & (\vec{r}_4 \times)\, R_4^0 & (\vec{r}_5 \times)\, R_5^0 \end{bmatrix} \begin{bmatrix} \overrightarrow{F_3^0} \\ \overrightarrow{F_4^0} \\ \overrightarrow{F_5^0} \end{bmatrix} \tag{7.36}$$

where R_i^0 is the transformation matrix from the coordinate frame of link i to the coordinate frame of the base link, and $r_i \times$ is the skew symmetric matrix of r_i. Multiplying through the force vectors for each of the three propellers comprising the HM-UAV the following system is obtained:

$$\begin{bmatrix} \overrightarrow{F_b} \\ \vec{\tau}_b \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ |F_3|\sin(q_3) & 0 & 0 & -|F_4|\sin(q_4) & 0 & 0 & 0 & 0 \\ |F_3|\cos(q_3) & 0 & 0 & |F_4|\cos(q_4) & 0 & 0 & 0 & |F_5| \\ -|F_3|(r_{3,y}c_3 - r_{3,z}s_3) & & & |F_4|(r_{4,z}s_4 - r_{4,y}c_4) & & & |F_5|r_{5,y} & \\ |F_3|r_{3,x}c_3 & & & |F_4|r_{4,x}c_4 & & & -|F_5|r_{5,x} & \\ -|F_3|r_{3,z}s_3 & & & -|F_4|r_{4,z}s_4 & & & 0 & \end{bmatrix} \tag{7.37}$$

$$\begin{bmatrix} \overrightarrow{F_b} \\ \vec{\tau}_b \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ |F_3|\sin(q_3) & -|F_4|\sin(q_4) & 0 \\ |F_3|\cos(q_3) & |F_4|\cos(q_4) & |F_5| \\ -|F_3|(r_{3,y}c_3 - r_{3,z}s_3) & |F_4|(r_{4,z}s_4 - r_{4,y}c_4) & |F_5|r_{5,y} \\ |F_3|r_{3,x}c_3 & |F_4|r_{4,x}c_4 & -|F_5|r_{5,x} \\ -|F_3|r_{3,z}s_3 & -|F_4|r_{4,z}s_4 & 0 \end{bmatrix} \tag{7.38}$$

Equation 7.38 comprises six equations, three for force, and three for torque, with five unknown control outputs, q_3, q_4 (the tilt angle of the right/left propeller), F_3, F_4 and F_5 (the thrust forces to be produced by the right, left, and tail propellers, respectively). Since the UAV cannot produce any thrust in the x_0 direction the first equation comprising Eqn 7.38 (row 1 of Equation 7.37) reduces to zero: $F_x^0 = 0$. This fact reduces the system of equations to five equations with five unknowns. Based on the desired generalized force given by the controller in Σ_0, the desired control outputs for the HM-UAV can be solved.

It is assumed that the angle of the two main tilting propellers (q_3 and q_4) can respond quickly relative to the frequency of the controller. It is also assumed that the UAV is in near hover conditions (i.e., small pitch and roll angles). Another assumption for the controller is that the tail propeller produces minimal thrust compared to the two main rotors, and therefore it is

assumed negligible when calculating the desired lift force F_z^0 values. This is also valid in the practical sense as the tail propeller's function is inducing pitch on the UAV and not lift (assuming there is no translational tendency as the HM-UAV pitches similar to the lateral translational tendency seen in traditional helicopters). As a result, the thrust for each of the tilting main propellers F_i ($i = 3$ and 4) as the propellers tilt can be split into the longitudinal and vertical (lift) components $F_{i_y}^0$, $F_{i_z}^0$ respectively as indicated in Equations 7.34 and 7.35.

$$F_{3_y}^0 = |F_3| \cos(q_3) \tag{7.39}$$

$$F_{4_y}^0 = |F_4| \cos(q_4) \tag{7.40}$$

$$F_{3_z}^0 = |F_3| \sin(q_3) \tag{7.41}$$

$$F_{4_z}^0 = |F_4| \sin(q_4) \tag{7.42}$$

To simplify the solution to the inverse dynamics for the UAV, it is assumed that the pitch control is performed primarily through the thrust generated by the rear (horizontal tail) propeller which induces a pitching moment ($\tau_{x_{des}}$) on the HM-UAM about its CoM. The corresponding force represented as $|F_5|$ can be computed as:

$$|F_5| = \frac{\tau_{x_{des}}}{r_{5,y}} \tag{7.43}$$

where $r_{5,y}$ is the horizontal distance (along the longitudinal axis of the HM-UAV) from the rear rotor to the origin of Σ_0 (Fig 7.5). The couple roll moment about the y_0 axis of the HM-UAV denoted as τ_y, is created by the difference in thrust between $F_{3_z}^0$ and $F_{4_z}^0$. The desired difference in thrust $F_{offset_{roll}}$ that will create the desired roll on the UAV is given by Eqn 7.39.

$$F_{offset_{roll}} = \frac{\tau_{y_{des}}}{r_{4_x}} \tag{7.44}$$

where r_{4_x} (Figure 7.5) is the x_0 component of the vector that extends from the rotor's axis of rotation to the origin of Σ_0.

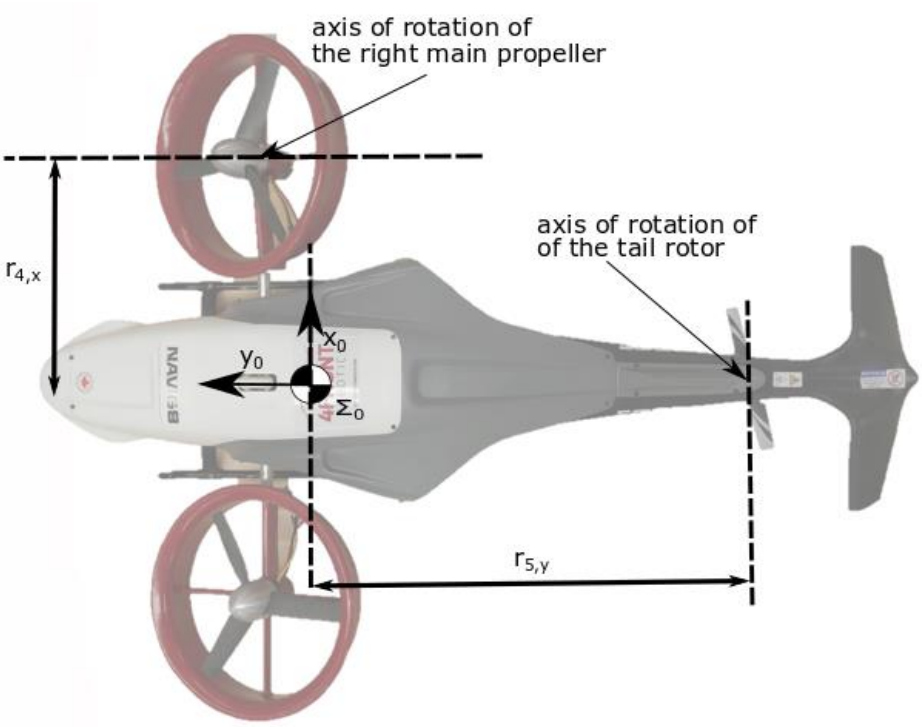


Figure 7.5: Propeller position vectors.

Similarly, the couple yaw moment about the aircraft's z_0 axis τ_z created mainly by the difference tilting angle between the right and left main propellers, $F_{3_y}^0$ and $F_{4_y}^0$, can be computed. The difference in F_y between the propellers, $F_{offset_{yaw}}$ is thus given by the following expression:

$$F_{offset_{yaw}} = \frac{\tau_{z_{des}}}{r_{4_x}} \tag{7.45}$$

The components of thrust generated by the main tilting rotors ($F_{3_z}^0$, $F_{4_z}^0$, $F_{3_y}^0$, $F_{4_y}^0$) can be found using the desired thrust and the couple moment offset values for roll and yaw. The obtained propeller thrust can be computed via Eqn 6.41 to Eqn 6.44.

$$F_{3_z}^0 = \frac{1}{2}\left(F_{z_{des}}^0 + F_{offset_{roll}} - F_5^0\right) \tag{7.46}$$

$$F_{4_z}^0 = \frac{1}{2}\left(F_{z_{des}}^0 - F_{offset_{roll}} - F_5^0\right) \tag{7.47}$$

$$F_{3_y}^0 = \frac{1}{2}\left(F_{y_{des}}^0 - F_{offset_{yaw}}\right) \tag{7.48}$$

$$F_{4_y}^0 = \frac{1}{2}\left(F_{y_{des}}^0 + F_{offset_{yaw}}\right) \tag{7.49}$$

From the values of the components of thrust force from the rotors, the desired magnitude of thrust and the desired propeller tilt angles (for the main propellers) can be derived. The magnitude of thrust for rotor i based on the component values is written as:

$$|F_i| = \sqrt{F_{i_y}^{0\,2} + F_{i_z}^{0\,2}} \tag{7.50}$$

The desired joint angle for rotor i based on the desired component values of thrust defined in Σ_i ($i = 3$ and 4) can be found using Equation 7.51:

$$q_i = \text{atan}\frac{F_{i_z}^i}{F_{i_y}^i} \tag{7.51}$$

7.5 Conclusion

The inverse kinematics for the HM-UAM were found in the first section of this chapter. This involved developing an algorithm to determine the combined motion of the HM-UAV and the robotic arm to follow trajectories that are outside of the workspace of the robotic arm. The algorithm also determined the desired position of the end-effector while the trajectory is outside of the workspace of the arm. In the second section, the control law for the HM-UAM was defined. The proposed/suggested adaptive sliding mode controller and the sliding surface was based on a PID controller. The inverse dynamics for the desired thrust for each of the UAV's thrusters under certain conditions and constraints was also derived.

Chapter 8 - Results

8.1 Introduction

In this chapter the simulation set-up, and the test results for various simulated experiments that were performed to analyze and validate the developed controller. Section 8.2 describes the set-up architecture of the simulation environment and provides information about the simulated and experimental parameters. Section 8.3 presents the results for the aerial end-effector trajectory tracking for various 2D and 3D trajectories, as well as using no loads or unknown loads on the end-effector.

8.2 Simulation Set-up

The simulation system that was set-up to test the developed HM-UAM controller employed the Coppelia Robotics V-REP software which is a commercial software used to simulate robotic as well as other systems. For this work, version 3.4.0 version 1 of the V-REP simulator used was used. Figure 8.1 illustrates V-REP's user interface with the simulated HM-UAM.

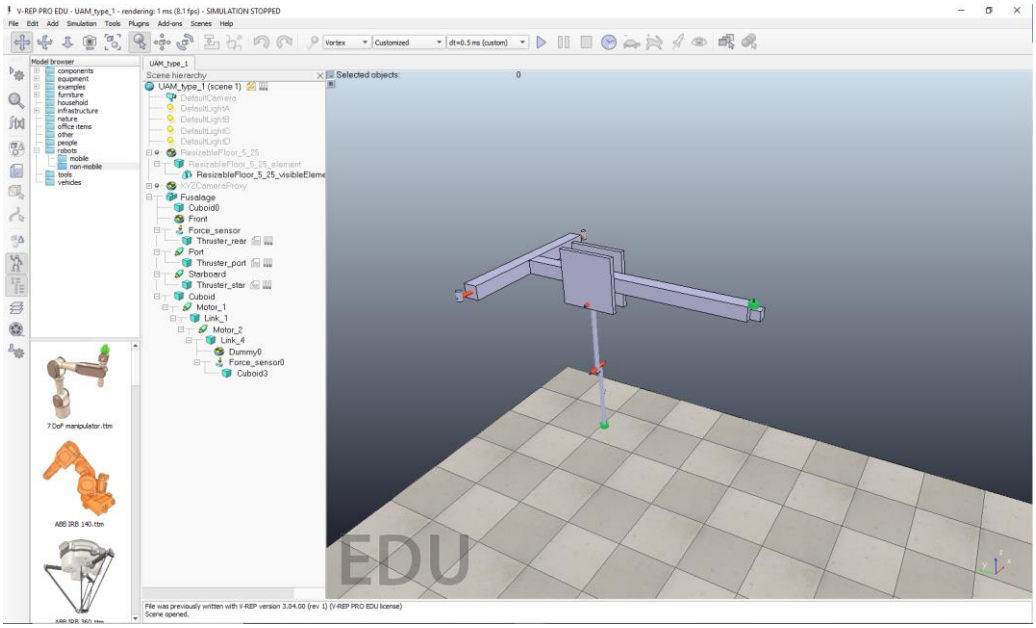


Figure 8.1: V-REP simulation.

Within V-REP a virtual HM-UAM was created containing all the corresponding linkages and joints. Subsequently, V-REP was interfaced with MATLAB through a remote API plugin which was used to run the developed controller while acquiring feedback information from V-REP's physical engine regarding the behavior of the HM-UAM. The controller and the simulation were run in synchronous operation, which means that the V-REP simulation would not advance to the next time step until the MATLAB script sent a trigger to advance the time step. Since the controller and V-REP simulation operate synchronously, the controller in MATLAB sends its commands to the HM-UAV at every time step. However, the desired controller frequency to be tested is less than the simulation frequency, so a zero order hold was used for the commands coming from the controller in the MATLAB script.

To complete the testing environment various simulation aspects parameters needed to be chosen to run the simulation. These included the simulation time step, the physics engine time step, and a suitable physics engine. The specific V-REP parameters and physics engine used in this work are shown in Table 8.1.

V-REP was selected due to the fact it is not a pure physics simulator, but a systems (kinematics) simulator that uses an independent physics engine to compute the dynamics. As a result, the kinematics simulation and the physics engine run at different frequencies. There are several available physics engines that are available to be used with V-REP such as ODE, Bullet, and Vortex. The chosen physics engine for this work was the commercially available software Vortex. Vortex was selected due to its closed source physics engine created by CM Labs Simulation which offers high fidelity, accurate results and is the most accurate physics engine available to be used with V-REP [52]. Once the simulation environment was set-up, the physical parameters for the HM-UAV and robotic arm such as mass, inertia, CoM, arm's link lengths,

maximum thrust output for the propellers, maximum torque output for the joints in the robot arm were defined.

Table 8.1: Simulation Parameters.

Parameter	Value
Simulation Time Step (s)	0.001
Physics Engine Time Step (s)	0.00025
Physics Engine	Vortex

The HM-UAM parameters used in this work were obtained directly from the CAD design files for the HM_UAV and the designed robot arm. Appendix A provides a complete list of all HM-UAM parameters used. A complete list of the controller parameters used during the simulations such as gain matrices Λ, K, and Γ required in the mathematical formulations are given in Appendix B.

8.3 Results

In order to test and analyze the controller in incremental levels of complexity a set of increasingly complex trajectories that the robot arm was commanded to track were selected. First a set of 2D trajectories are used to test the controller's ability to follow continuous smooth curves such as circular paths followed by trajectories having abrupt or sharper changes in direction such as square paths. Subsequently, the controller's ability to guide the HM-UAM to follow smooth trajectories with the end-effector under a priori unknown disturbances is then tested. For this the disturbance consisted of an unknown point mass attached to the end-effector. The purpose of this is to test the controller under typical conditions that any UAM would encounter in real world applications. Subsequently, the controller is tested using 3D smooth trajectories where the HM-UAV would be required to translate and/or rotate (yaw) to enable the arm's end-effector to track the curve under specific constraints such as speed. Finally, cooperatively trajectory tracking

where the arm follows a projected path onto its workspace while the HM-UAV rotates to bring the real path onto the arm's workspace is investigated.

8.3.1 Circular Trajectory

The HM-UAM controller's ability to follow smooth continuous curves with the end-effector is tested by following a 2D circle trajectory. This is a well-known reference trajectory common for 2-DOF revolute-revolute robot arms. For this simulation, the initial position of the HM-UAV's CoM was set to $X_G = [0 \quad 0 \quad 2.5]^T m$ which indicates a hovering UAV at 2.5 (m) above the ground surface. The initial position of the joints of the robotic arm were set to $\theta_1 = 0$ rad and $\theta_2 = 0$ rad. The trajectory of had a radius of 0.05m, which was followed at a frequency of 0.25 Hz. That is the arm was commanded to complete its circular path in 4.0 seconds. The center of the circular path was positioned at $[x_g, y_g, z_g] = [0, 0.25, 1.875]$ (m). The circular path was entirely inside the longitudinal plane of the HM-UAM and within the robot arm's workspace. Figure 8.2 shows a schematic representation of the test and the location of the circular path to be followed w.r.t the HM-UAV and the robot arm.

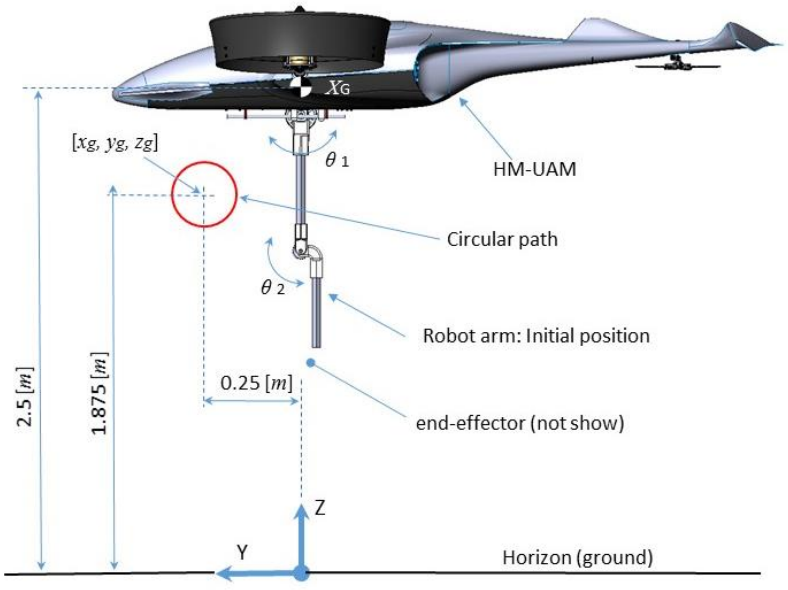


Figure 8.2: Circular path test.

During the test, the HM-UAV was always commanded to perform a typical (horizontal) hover maneuver. Figure 8.3 shows the results for the corresponding tracking.

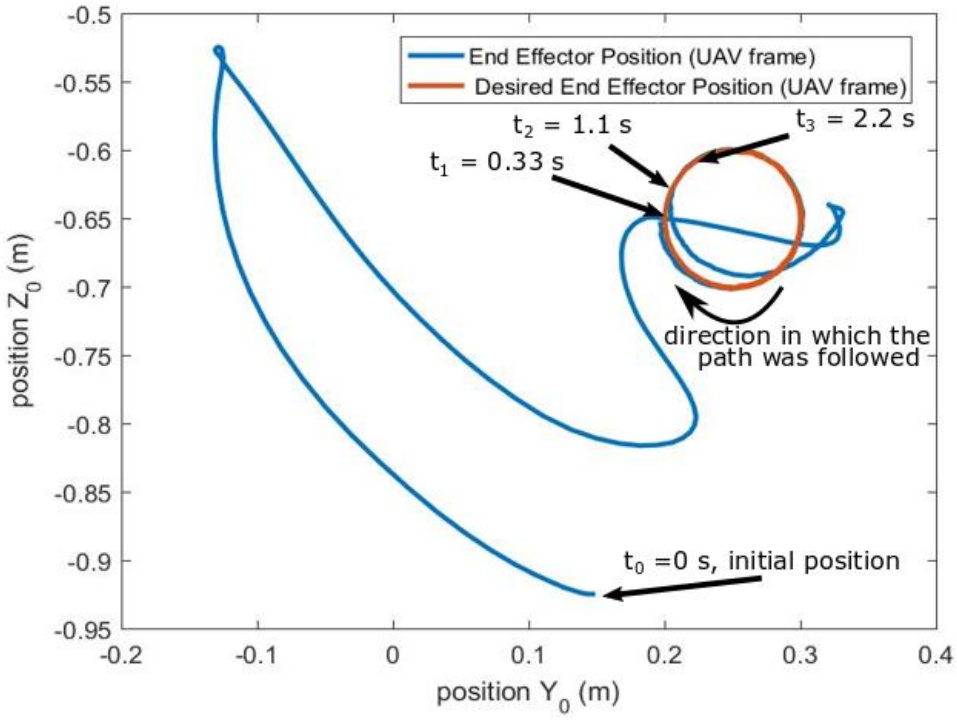


Figure 8.3: End-effector circular path trajectory following.

As the robot arm attempts to intercept and track the circular path trajectory with the its end-effector, both of the joints in the robot arm attempt to reach their desired joint positions and velocities. Since the initial tracking error is large, the joints attempt to rapidly intercept the desired trajectory. However, the desired control effort exceeds the torque in which the joints actuators are capable of producing and the actuators become saturated. This leads to the two joints in the robot arm oscillating about their desired joint positions (the joints oscillating about their desired joint positions can be seen in Figures 8.4 and 8.5) creating the smooth overshooting paths shown in Figure 8.3. The control effort is focused on overcoming the inertial forces acting on each of the links of the robotic arm, as well as other forces such as gravity, while attempting

to bring each of the links to their desired state. As the joints converge with the desired trajectory, the tracking error is reduced, and the inertial forces are reduced, and the desired control effort no longer saturates the actuators (Figure 8.7).

From the results, it can be seen that the arm takes 1.0 seconds to fully converge and track the path with very little error. In Figure 8.4, the total position error of the end effector is given.

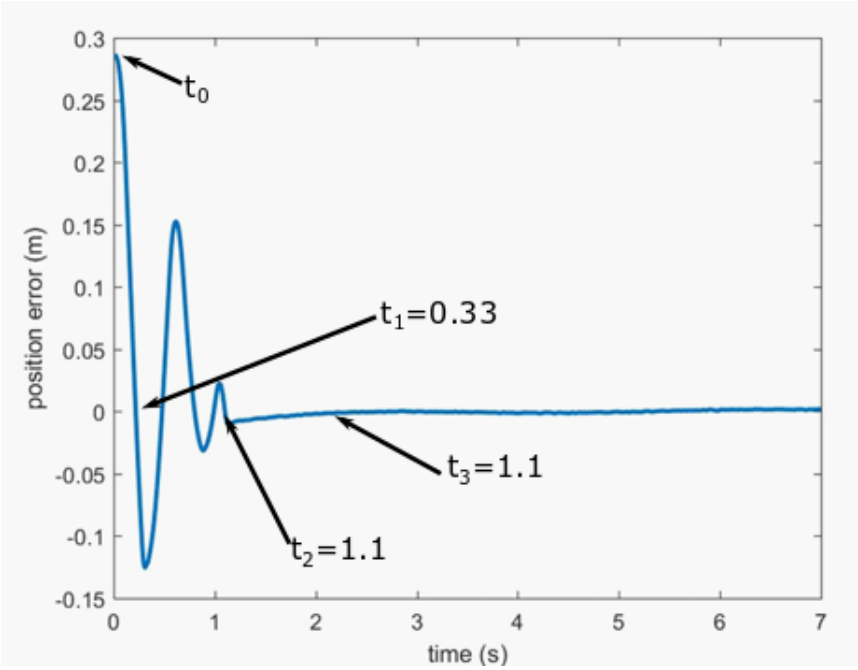


Figure 8.4: End Effector Circle Trajectory Following.

The end-effector error is not the same as the joint following error in the robot arm. Figure 8.5, and Figure 8.6 shows the trajectory following for the joints in the robot arm. Figure 8.7 shows the control effort (Joint Torque).

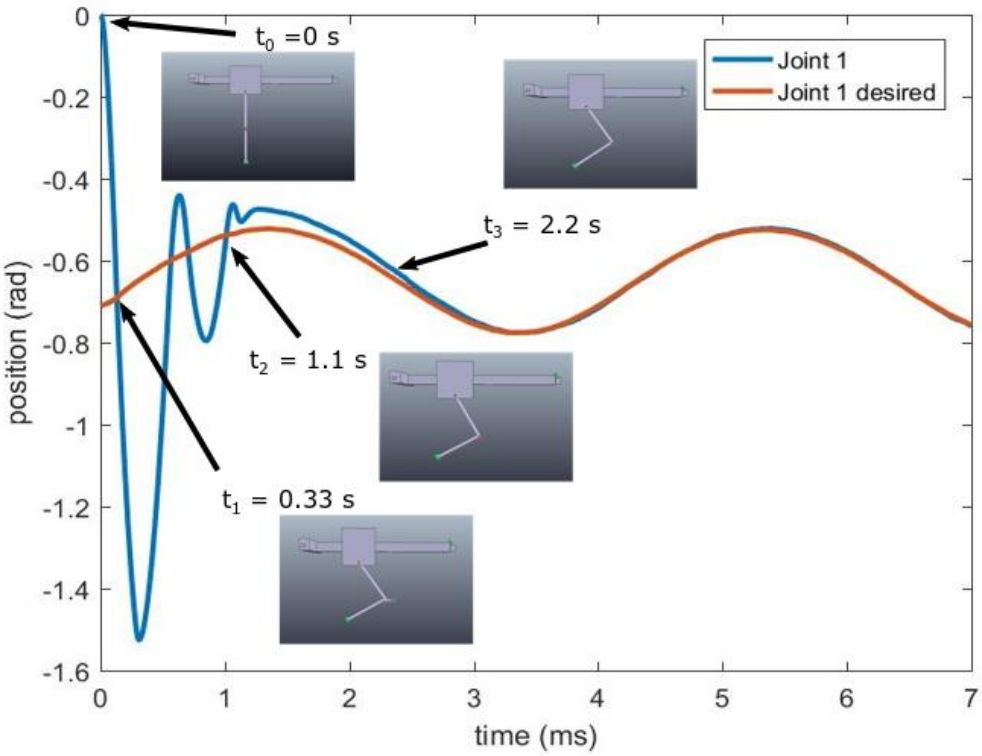


Figure 8.5: Robot arm joint 1 trajectory following.

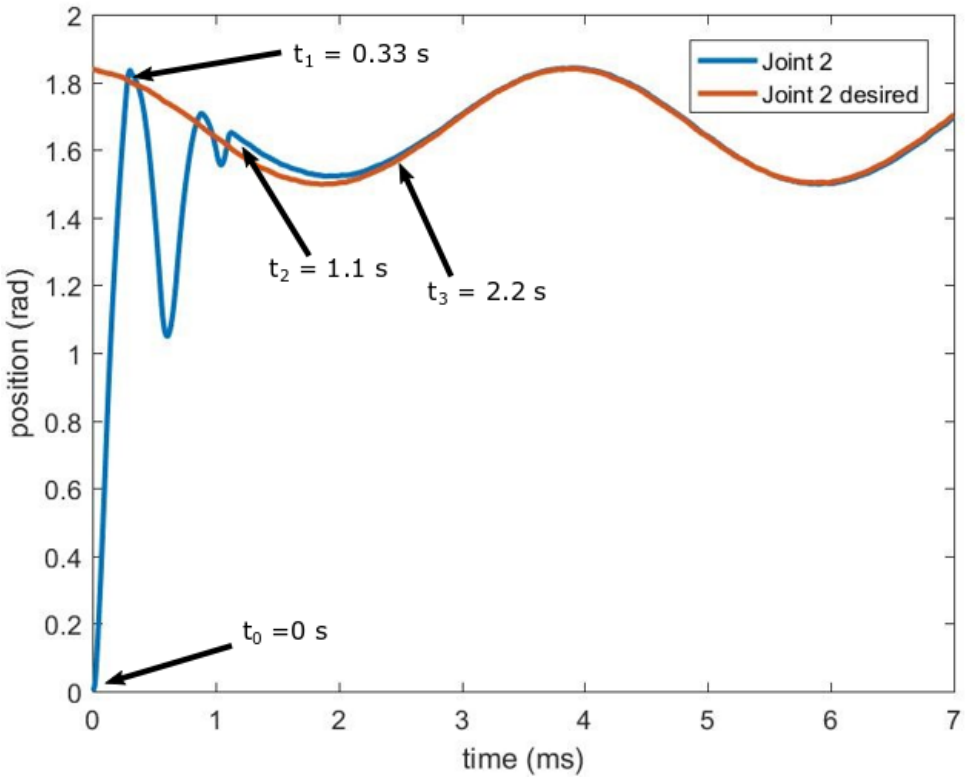


Figure 8.6: Robot arm Joint 2 trajectory following.

From Figures 8.5 and 8.6 it is seen that once the HM-UAM converges to effectively tracking the path it is able to continue doing so for 1, 2, and more cycles.

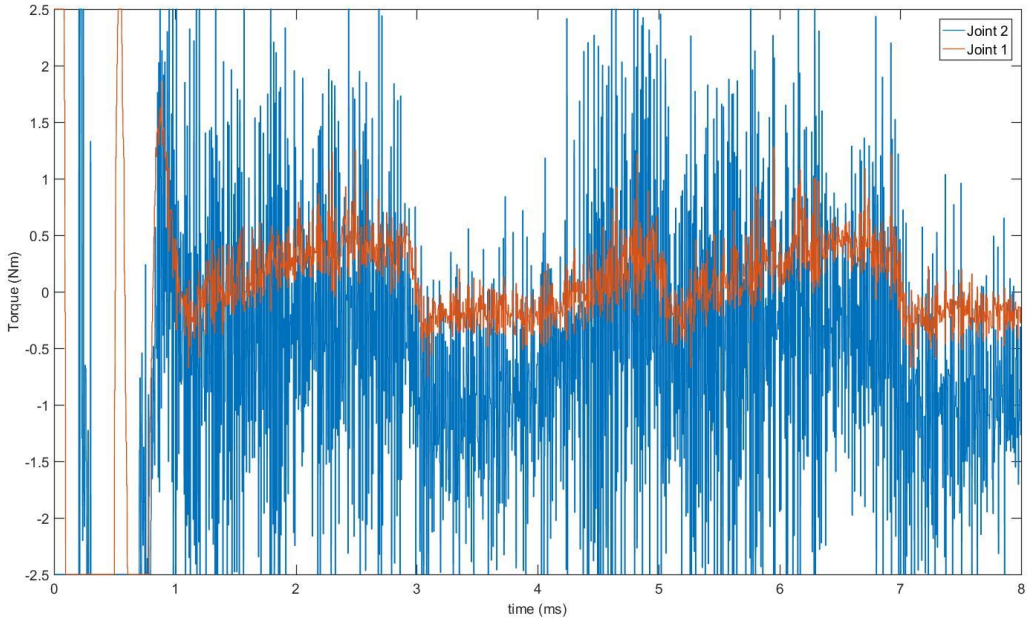


Figure 8.7: Robotic arm joint torques.

8.3.2 Square Trajectory

The ability for the HM-UAM controller to follow end-effector trajectories with discontinuous changes in curvature is tested by commanding the HM-UAM to follow a square trajectory. The desired trajectory was set-up so that the desired path along the perimeter of the square did not begin until 1.5 seconds into the simulation. This was done to ensure that the end-effector would begin the tracking of the square at its desired position. For this case, the end-effector was initially positioned at ($x_0 = 0.003$, $y_0 = 0.2002$, $z_0 = 1.5750$) and its desired start position of the trajectory was ($x_G = 0$ m, $y_g = 0$ m, $z_G = 1.75$). The desired square trajectory had the following dimensions 0.15 m x 0.15 m and its center of geometry was located at ($x_G = 0$ m, $y_g = 0.075$ m, $z_G = 1.825$) m as shown in Figure 8.8. Similar to the circular path the square trajectory was

commanded to be followed at a frequency of 0.167 Hz. In this case the arm was commanded to complete the path in 6 seconds.

The total trajectory following error for the end-effector is shown in Figure 8.9. As seen from Figure 8.8, the end-effector is able to closely track the desired trajectory and from Figure 8.8 it can be seen that the trajectory following error is minimal after 1.5 seconds.

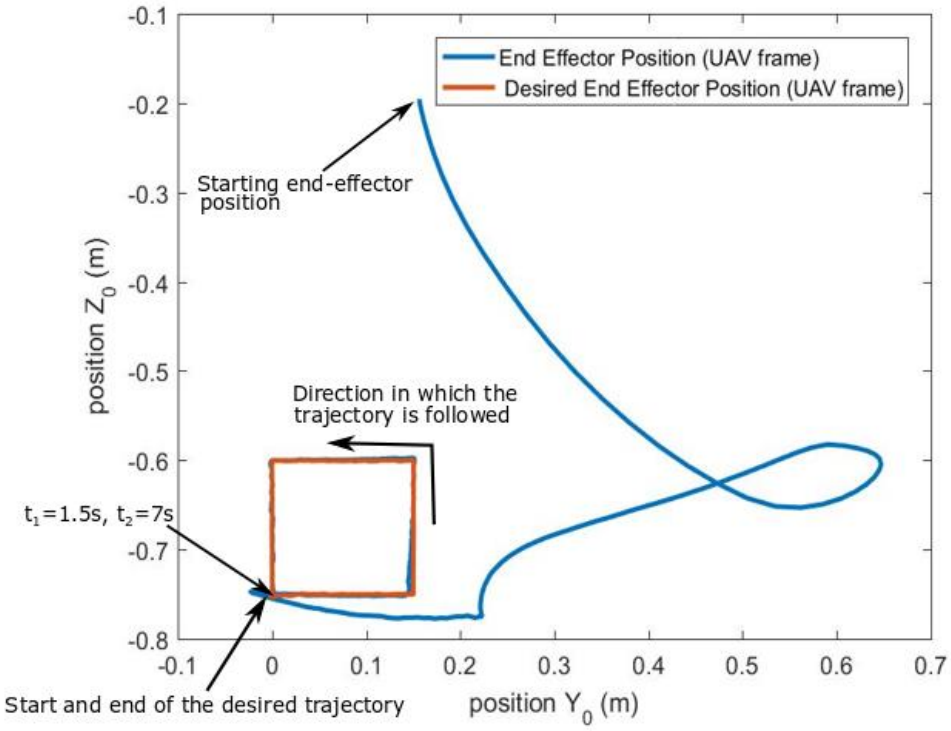


Figure 8.8: End Effector Circle Trajectory Following.

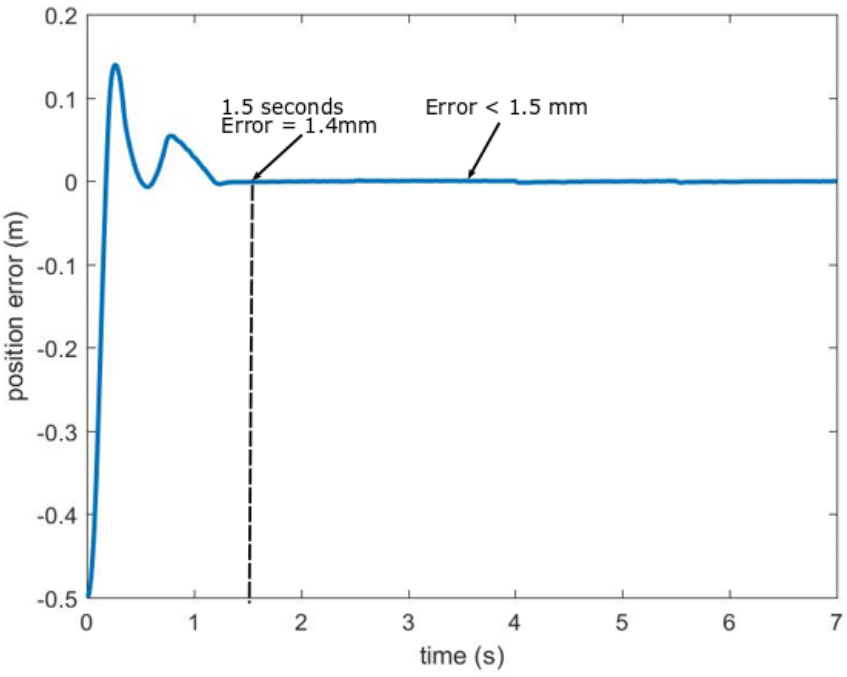


Figure 8.9: End-effector error.

The end-effector completed one successful pass around the desired square trajectory in the experiment. This test was repeated several times to ensure the accuracy of results. The direction and the speed in which the end-effector tracked the desired trajectory was part of the control requirements and the controller was successful in its trajectory tracking attempt.

8.3.3 Square Trajectory with Unknown Load

The ability for the controller to adapt to external disturbances is tested by adding an unmodeled load of 0.4 kg to the end-effector of the HM-UAM prior to starting its square reference trajectory following. The reference trajectory followed by the end-effector is a square trajectory shown in Figure 8.10 having the following dimensions 0.25 m x 0.25 m and with its center at ($x_G = 0$ m, $y_g = 0.225$ m, $z_G = 1.85$). The blue line in Figure 8.10 is the actual path followed by the end-effector. The total end-effector trajectory error is shown in Figure 8.11. The time-stamps (e.g, $t_0, \ldots t_4$) shown in Figure 8.11 correspond to the same time stamps in Figure

8.10. At time t_3 the arm assumes the end-effector has reached the top left vertex of the trajectory and starts following the top horizontal path. As a result, the error difference between t_3 and t_4 (Figure 8.11) changes rapidly but it doesn't mean that the end-effector has performed an aggressive change in trajectory.

It is seen that the trajectory error is much higher when there is a relatively large unknow load. The differences in the ability of the arm to follow the trajectory for the different sides of the arm is because of different dynamic and static loading based on the robot arm's joint positions. However, this is not observed at the bottom of the trajectory (between times t_1 and t_2) when the end-effector is further from the HM-UAV's fuselage and the arm is closer to vertical requiring less joint effort to follow the trajectory (and the joint actuators are not saturated). As the trajectory approaches the HM-UAV's fuselage, the more control effort required and the joint actuators become saturated, hence the larger following error at the top of the square in Figure 8.10.

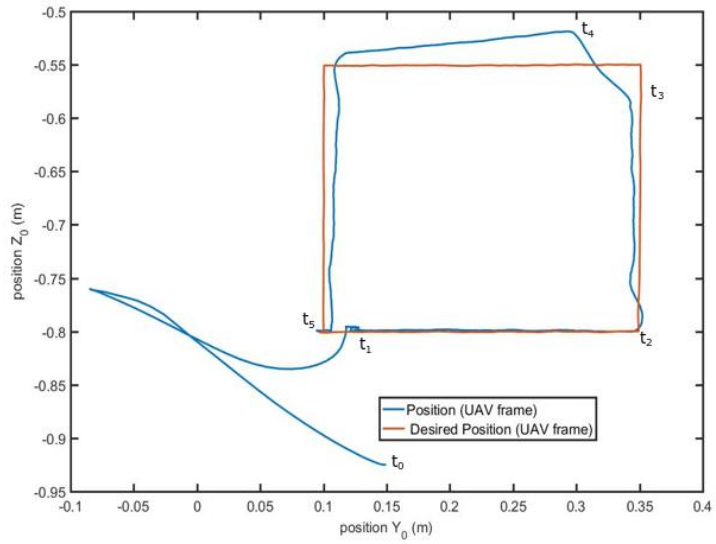


Figure 8.10: Square End-Effector trajectory following with unknown load.

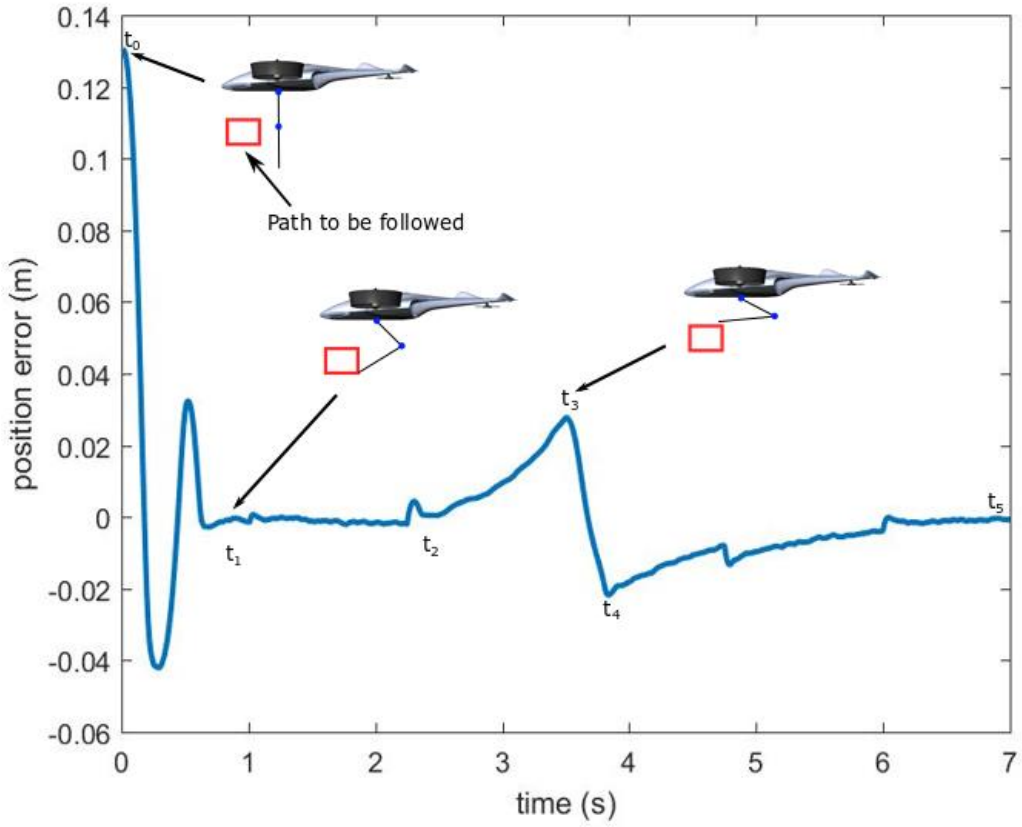


Figure 8.11: End effector following error under an unknown load.

Other similar tests were performed following 2D trajectories while loading the arm with varies loads. In all such tests, similar results described above were observed. It must be pointed out that at this time the motion of the HM-UAV was considered near perfect and the errors reported are due to the arm only. In real applications, the HM-UAV would also contribute in some was to the error but techniques could be developed to have the HM-UAV assist the arm the arm during the trajectory following (translate the aircraft).

8.3.4 Projected Circle Trajectory

The desired end-effector's trajectory is a circular trajectory that is positioned in the $Y_G Z_G$ plane while the HM-UAV follows a yaw trajectory taking the arm away from the trajectory. As a

result, as the HM-UAV yaws, the desired trajectory for the end-effector leaves the plane in which the robot arm operates in, and the desired trajectory is projected into the plane of the arm.

A set of tests was conducted to analyze the cooperative behaviour of the HM-UAV and arm while following 2D paths that were outside the workspace of the robot arm. In this set of tests, the HM-UAV was commanded to yaw taking the arm's workspace plane away or closer to the plane of the trajectory.

Simultaneously, (as the HM-UAV yaws) the end-effector was targeted to follow a projected virtual path that was generated by projecting the (static) real path into the workspace of the arm is illustrated in Figure 8.12. The result of the continuous motion of the HM-UAV is a dynamic projected path which changes its shape (geometry) continuously in the workspace plane of the arm as illustrated in Figure 8.12 for three timesteps.

For such test attention was placed to not yaw the HM-UAV too much to keep the projected path always within the arm's reach (otherwise the HM-UAV would be required to translate depending on the current state of the arm). Figure 8.13 shows the yaw movement that the HM-UAV was commanded to follow (i.e., from 0 to 12 degrees) which the aircraft followed very closely.

As seen from Figure 8.12 the end-effector trajectory within the plane of the arm changes as the HM-UAV yaws. This results in the change in the desired circular path (the red line) in Figure 8.15, where it can be seen that the shape and radius of the circular path changes and that the arm follows this projected path. In Figure 8.15, Loop 1 is the circular path when the yaw of the UAV is minimal. As the simulation progresses, the circular path loops back around and it can be seen from Loop 2 that the radius of the circular path has increased. Figure 8.14 shows the end-effector tracking split up into its y_G and z_G components. In Figure 8.14 it is seen that the arm

takes longer to settle and follow the y_G component of the trajectory compared to the z_G component because the y_G is the only part of the trajectory that is distorted by the yawing of the HM-UAV.

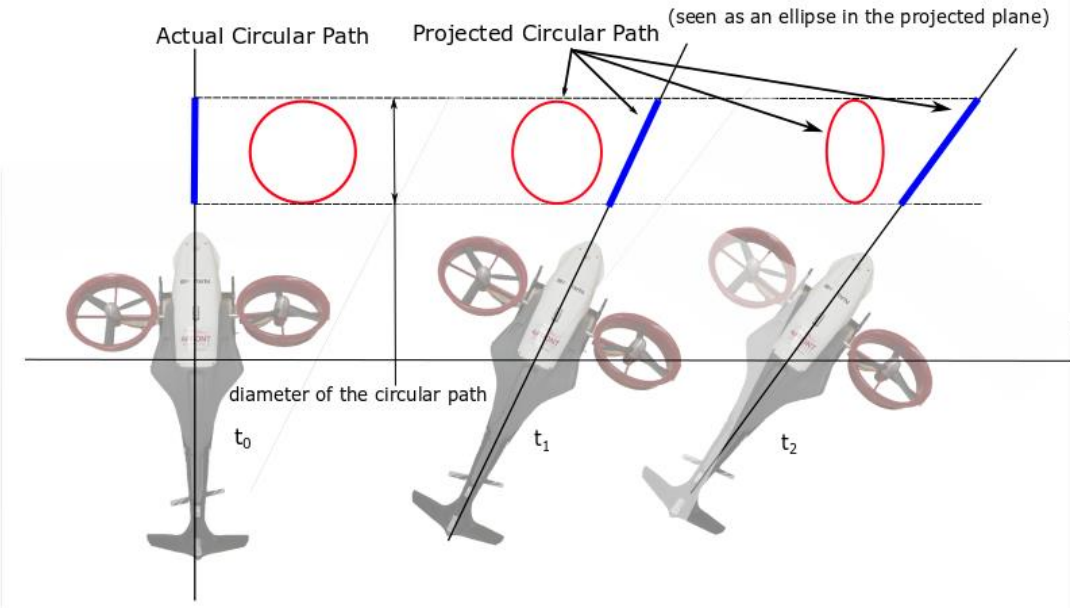


Figure 8.12: Projected Circular Path for Yawed HM-UAV.

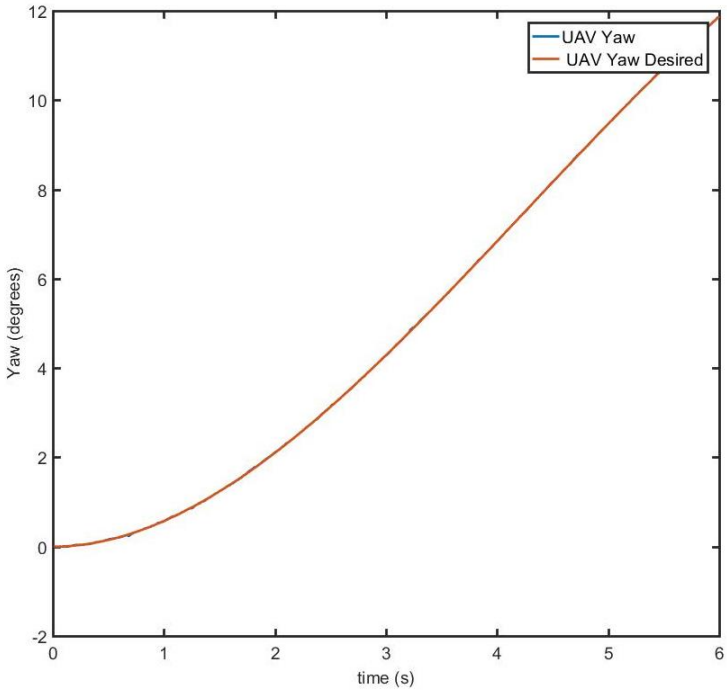


Figure 8.13: Yaw Trajectory for UAV.

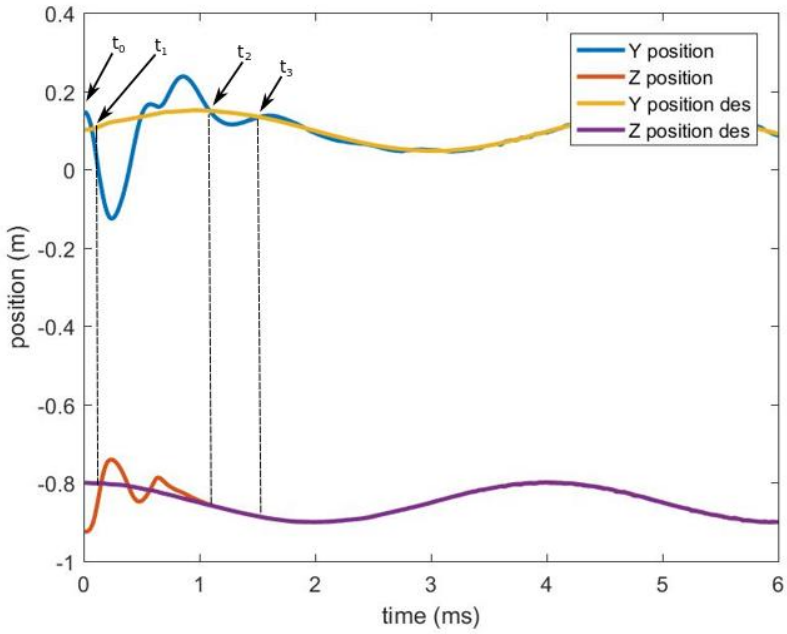


Figure 8.14: End-effector following a circular trajectory projected into the arm's planar workspace.

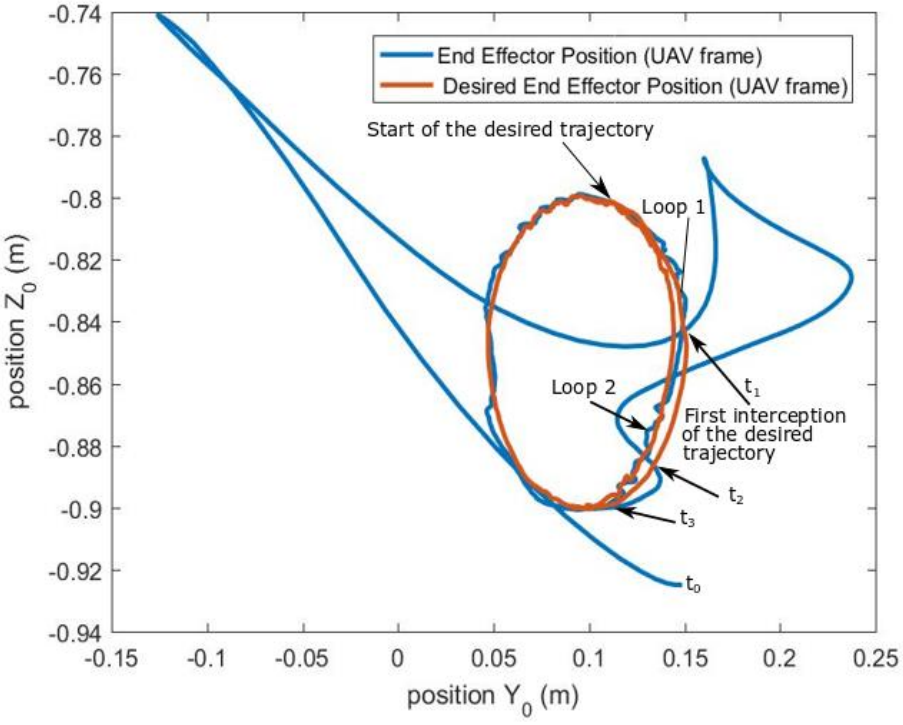


Figure 8.15 Circle Trajectory following a circular trajectory projected into the HM-UAV frame.

Figure 8.16 shows the trajectory following error for following a circular path projected into the frame of the arm. After 1.5 seconds into the simulation, the end-effector following error doesn't rise above 2.5 mm for the rest of the simulation.

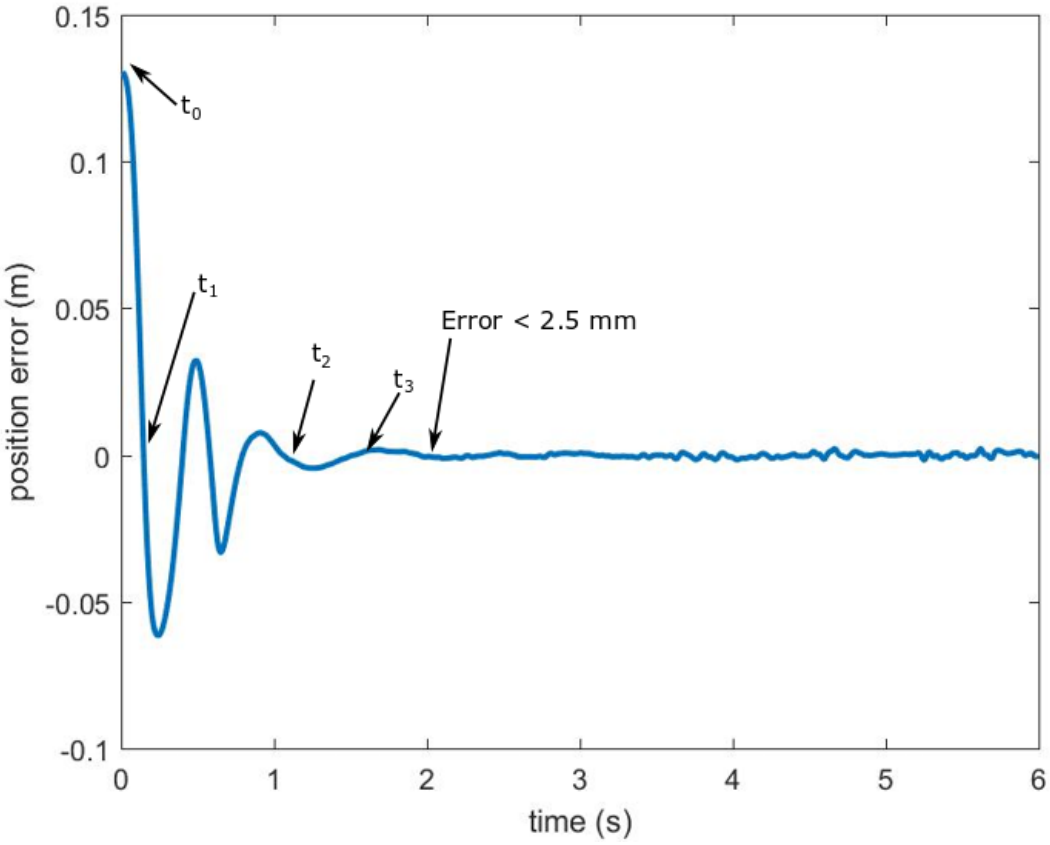


Figure 8.16: End-effector trajectory error for following a circular trajectory projected into the HM-UAV frame.

8.3.4 3D Trajectory

The finale set of tests performed include providing a smooth 3D trajectory for the end-effector to follow. In contrast with the previous tests where the HM-UAV was commanded to move the arm away from the desired trajectory, in this case, the HM-UAV yaws to keep the plane of the arm near the desired trajectory as it progresses through 3D space. Similar to the

previous tests, here the HM-UAV was not required to translate while moving, enabling the end-effector to follow the real trajectory via a projected path into the plane of the arm. In ideal conditions, the current state of the end-effector would always be in contact with the real path. That is, the projected point (where the end-effector is) coincides with the real trajectory point while the remaining projected points (within the arm's workspace plane) are virtual path points. In reality, however, depending on the complexity of the curve to be followed, the capabilities of the HM-UAM, ect. the end-effector might not always be precisely positioned on the real curve at all times. In the case that the trajectory is not within the plane of the arm, the end-effector will then follow the projected trajectory into the workspace of the arm.

For this test, the 3D curve used is illustrated in Figure 8.17. The CoM of HM-UAV (and thus the HM-UAV) is positioned at coordinates ($x_G = 0$ m, $y_g = 0$ m, $z_G = 2.50$ m) (not shown in Figure 8.17). The desired trajectory for the arm in 3D space is created by first creating a desired commanded yaw trajectory (Eqn 8.1) for the UAV. Then y_G (Eqn 8.2), and z_g (Eqn 8.3) command trajectories are then projected into the desired longitudinal plane of the HM-UAV and the resultant 3D trajectory is such that it always lies within the workspace of the arm provided the HM-UAV yaws (no need to translate) is used.

$$Yaw_{des} = \frac{\pi}{4}\sin\left(t * \frac{2}{3}\right) \tag{8.1}$$

$$y_{G_{des}} = \sin(t * 2\pi * 0.25) * 0.2 + 0.3 \tag{8.2}$$

$$z_{G_{des}} = \sin(t * 2\pi * 0.25) * 0.1 + 1.7 \tag{8.3}$$

It can be seen from Figure 8.17, that the end-effector closely follows the desired 3D trajectory after the end-effector and HM-UAV motion combination reaches the trajectory and settles. The end-effector trajectory following error for the projected trajectory is shown in Figure 8.18 where it can be seen that the end-effector and HM-UAV converges in 1.4 seconds (less than

1.5 mm of trajectory following error for the rest of the simulation) with the projected trajectory. The desired trajectory for the end-effector caused the HM-UAV to follow an aggressive yaw yet smooth trajectory as shown in Figure 8.19, where it can be seen that while the HM-UAV lags behind the desired trajectory the lag is within acceptable limits.

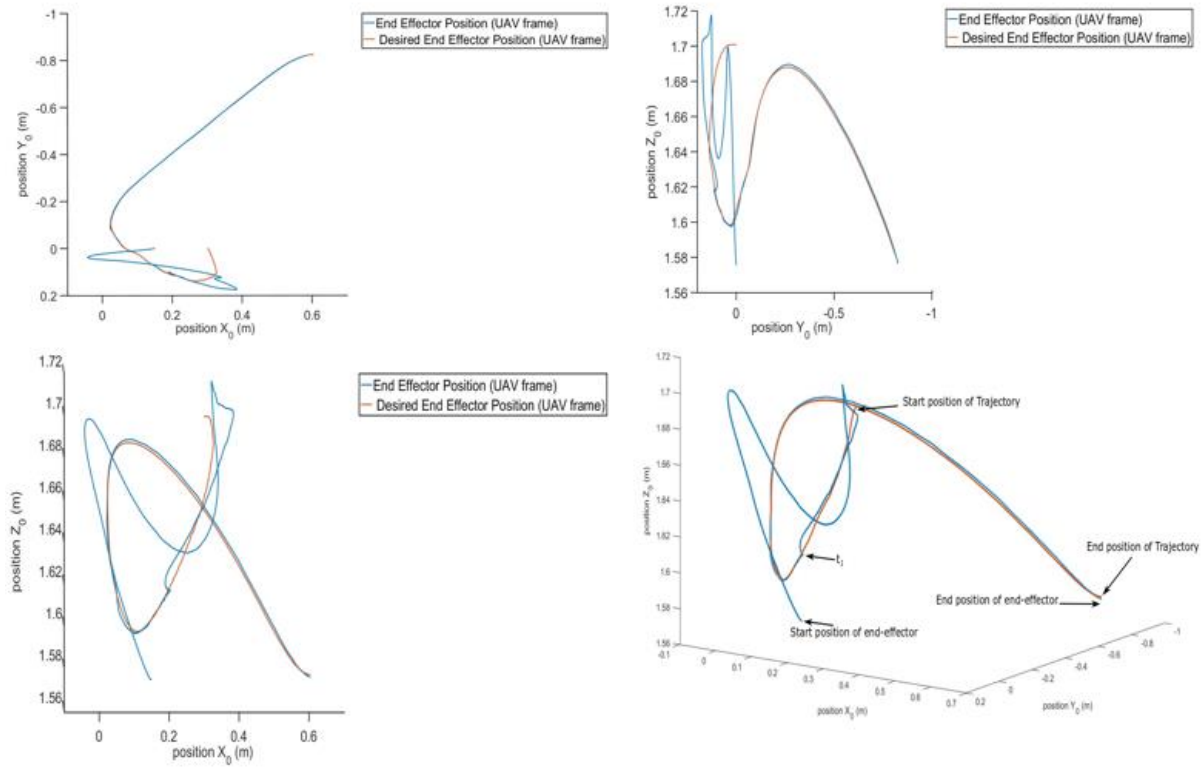


Figure 8.17: 3D end-effector trajectory following.

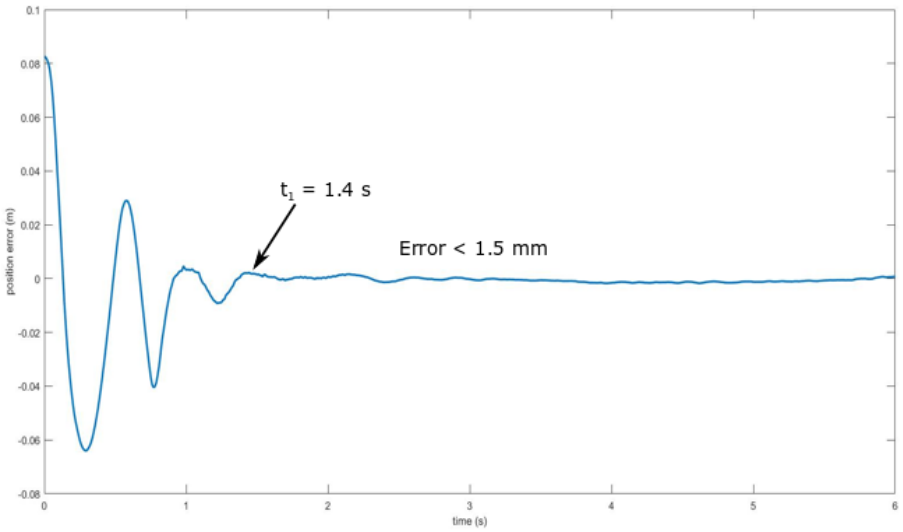


Figure 8.18: End-effector trajectory following error.

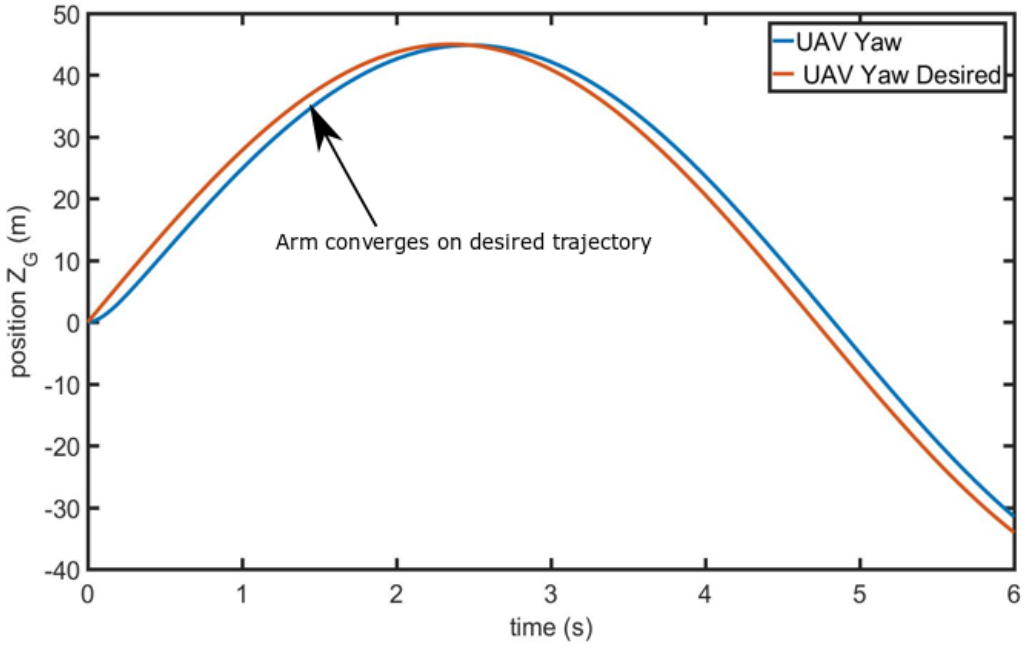


Figure 8.19: HM-UAM yaw trajectory following.

8.4 Summary

The above simulation results and analysis show the capabilities of the proposed control architecture for the HM-UAM. The HM-UAM controller was subjected to follow various end-effector trajectories to test different aspects of the controller. The ability of the controller to

follow smooth continuous curves was tested successfully by following diverse circular trajectories. A set of square trajectories were used to test the controller's ability to enable the end-effector to follow trajectories where the curvature of the path changes abruptly. From the result, it was shown that the HM-UAM is able to follow the desired trajectory closely but required the cooperative effort of the HM-UAV and the arm. The controller's ability to adapt was then tested by adding a unmodeled mass to the end-effector of the arm. While not performing as well as the case where no added mass was present, the arm was still able to follow the trajectory to a satisfactory level. Next, the robot arm was tasked with following a 2D circle trajectory while the HM-UAV is rotated. The projected 2D circular trajectory into the plane of the robot arm, was shown to be a successful approach. The final test of the HM-UAM controller included following 3D end-effector trajectories where coordination between the arm and the HM-UAV was required. The HM-UAV was shown to be able to follow the required yaw trajectory while the robot arm was able to follow the projected trajectory with its end-effector with minimal error.

Chapter 9 - Conclusion

This thesis addresses the problem of the design and implementation of a controller capable of tracking a reference trajectory in 3D space for a highly maneuverable thrust vectoring bi-copter equipped with a pitch trim horizontal tail rotor and a 2-DoF revolute-revolute planar arm under an unknown load attached to the end effector. Using the sequence of steps outlined in this thesis (Chapter 3), a control algorithm for the HM-UAM was developed to enable the end-effector of the attached robot arm to effectively follow trajectories in 3D space. First a mathematical frame work was developed to model the coupled kinematics and dynamics of HM-UAV and robot arm to be used within the control algorithm. Next the implementation of an adaptive sliding mode controller for the HM-UAM is performed. The implemented controller is then tested within an simulation environment. Various 2D reference trajectories were followed with the end-effector of the robot successfully demonstrating the ability of the control algorithm for the HM-UAV and robot arm to work successfully despite the coupled dynamic interactions between the HM-UAV and the robot arm. Next, the controller for the HM-UAM is tested by following a reference trajectory with the end-effector while adapting to an unmodeled mass added to the end-effector. Finally, the HM-UAM controller is tested successfully by following a 3D reference trajectory with minimal error.

The contributions proposed in Chapter 3 are achieved successfully in this thesis. Future works however, still remain to be performed.

Future works can be divided into both long term and short-term activities. Future short-term activities include: i) Experimental testing of the robustness of the proposed algorithm by introducing sensory error ii) expanding the proposed control algorithm to include grippers to

perform aerial grasping and manipulation maneuvers, and iii) expanding the controller to deal with disturbances such as flexible links in the robot arm.

The proposed long-term future research is envisioned to: i) Extend the proposed controller for the HM-UAM to work in cooperation with other HM-UAMs to grasp and manipulate objects, ii) extend the capabilities of the controller to grasp long and flexible objects, and iii) Extending the control algorithm for the HM-UAM to perform aerial grasping and manipulation maneuvers in confined spaces.

Appendix A: HM-UAM Parameters

UAV Parameters

Mass of the UAV: 5 kg

Inertia Matrix of the UAV:

$$I_0 = \begin{bmatrix} 0.155 & 0 & 0 \\ 0 & 0.38 & 0 \\ 0 & 0 & 0.245 \end{bmatrix} m^2 kg$$

Link 1 Parameters

Mass of Link 1: 0.221 kg

Inertia Matrix of Link 1:

$$I_1 = \begin{bmatrix} 0.00397 & 0 & 0 \\ 0 & 0.00397 & 0 \\ 0 & 0 & 7.81 * 10^{-5} \end{bmatrix} m^2 kg$$

$$d_1 = \begin{bmatrix} 0.2 \\ 0 \\ 0 \end{bmatrix} m$$

$$p_1 = \begin{bmatrix} -0.05 \\ 0.2 \\ -0.125 \end{bmatrix} m$$

$$u_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Link 2 Parameters

Mass of Link 2: 0.221 kg

Inertia Matrix of Link 2:

$$I_2 = \begin{bmatrix} 0.00397 & 0 & 0 \\ 0 & 0.00397 & 0 \\ 0 & 0 & 7.81 * 10^{-5} \end{bmatrix} m^2 kg$$

$$d_2 = \begin{bmatrix} 0.2 \\ 0 \\ 0 \end{bmatrix} m$$

$$p_2 = \begin{bmatrix} 0.2 \\ 0 \\ 0 \end{bmatrix} m$$

$$u_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Link 3 Parameters

Mass of Link 3: 0.341 kg

Inertia Matrix of Link 3:

$$I_3 = \begin{bmatrix} 6.905 * 10^{-5} & 0 & 0 \\ 0 & 6.905 * 10^{-5} & 0 \\ 0 & 0 & 6.905 * 10^{-5} \end{bmatrix} m^2 kg$$

$$d_3 = \begin{bmatrix} -0.01 \\ 0 \\ 0.075 \end{bmatrix} m$$

$$p_3 = \begin{bmatrix} 0.425 \\ 0.62 \\ 2.5 \end{bmatrix} m, u_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Link 4 Parameters

Mass of Link 4: 0.341 kg

Inertia Matrix of Link 4:

$$I_4 = \begin{bmatrix} 6.905 * 10^{-5} & 0 & 0 \\ 0 & 6.905 * 10^{-5} & 0 \\ 0 & 0 & 6.905 * 10^{-5} \end{bmatrix} m^2 kg$$

$$d_4 = \begin{bmatrix} 0.01 \\ 0 \\ 0.075 \end{bmatrix} m$$

$$p_4 = \begin{bmatrix} -0.425 \\ 0.62 \\ 2.5 \end{bmatrix} m, u_4 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Appendix B: HM-UAM Controller Parameters

$$\Lambda = \begin{bmatrix} 20 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 22 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 50 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 50 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 15 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 60 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 60 \end{bmatrix}$$

$$K = \begin{bmatrix} 0.05 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 25 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 25 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 900 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 900 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 20 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 20 \end{bmatrix}$$

$$\Gamma = \begin{bmatrix} 0.001 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.0002 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0002 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2.0 \end{bmatrix}$$

References

[1] K. Herrick, "Development of the unmanned aerial vehicle market: forecasts and trends", Air & Space Europe, vol. 2, no. 2, pp. 25-27, 2000.

[2] Dyveke Weissbach, Kathryn Tebbe, (2016) "Drones in sight: rapid growth through M&A's in a soaring new industry", Strategic Direction, Vol. 32 Issue: 6, pp.37-39

[3] B. Canis, "Unmanned Aircraft Systems (UAS): Commercial Outlook for a New Industry", Congressional Research Service, Rep. R44192 7-5700 (1-2), September. 2015

[4] R. Finn and D. Wright, "Privacy, data protection and ethics for civil drone practice: A survey of industry, regulators and civil society organisations", Computer Law & Security Review, vol. 32, no. 4, pp. 577-586, 2016.

[5] B. Rao, A. Gopi and R. Maione, "The societal impact of commercial drones", Technology in Society, vol. 45, pp. 83-90, 2016.

[6] M. Boyle, "The Race for Drones", Orbis, vol. 59, no. 1, pp. 76-94, 2015.

[7] K. Alexis, G. Darivianakis, M. Burri and R. Siegwart, "Aerial robotic contact-based inspection: planning and control", Autonomous Robots, vol. 40, no. 4, pp. 631-655, 2015.

[8] Burri, M., Nikolic, J., Hurzeler, C., Caprari, G., & Siegwart, R. (2012). Aerial service robots for visual inspection of thermal power plant boiler systems. In 2012 2nd International conference on applied robotics for the power industry

[9] Metni, N., & Hamel, T. (2007). A UAV for bridge inspection: Visual servoing control law with orientation limits. Automation in Construction

[10] C. Thiels, J. Aho, S. Zietlow and D. Jenkins, "Use of Unmanned Aerial Vehicles for Medical Product Transport", Air Medical Journal, vol. 34, no. 2, pp. 104-108, 2015.

[11] E. Fumagalli, M. Bibuli, M. Caccia, E. Zereik, F. Bianco, L. Gasperini, G. Stanghellini and G. Bruzzone, "Combined Acoustic and Video Characterization of Coastal Environment by means of Unmanned Surface Vehicles", IFAC Proceedings Volumes, vol. 47, no. 3, pp. 4240-4245, 2014.

[12] Bhattacharjee, Dhrubajyoti, Unmanned Aerial Vehicles and Counter Terrorism Operations (April 30, 2015). Available at SSRN: https://ssrn.com/abstract=2608969 or http://dx.doi.org/10.2139/ssrn.2608969

[13] Floreano, D., & Wood, R. J. (2015). Science, technology and the future of small autonomous drones. Nature, 521(7553), 460-466. Retrieved from http://ezproxy.lib.ucalgary.ca/login?url=https://search.proquest.com/docview/1685003576?accountid=9838

[14] C. Gomez and H. Purdie, "UAV- based Photogrammetry and Geocomputing for Hazards and Disaster Risk Monitoring – A Review", Geoenvironmental Disasters, vol. 3, no. 1, 2016.

[15] L. J. Love, J. F. Jansen, and F. G. Pin, Compensation of Wave-Induced Motion and Force Phenomena for Ship-Based High Performance Robotic and Human Amplifying Systems Prepared by, no. October. 2003.

[16] H. Tsukagoshi, M. Watanabe, T. Hamada, D. Ashlih, and R. Iizuka, "Aerial Manipulator with Perching and Door-opening Capability," *2015 IEEE Int. Conf. Robot. Autom.*, pp. 4663–4668, 2015.

[17] Jiuh-Biing Sheu, "An emergency logistics distribution approach for quick response to urgent relief demand in disasters", In Transportation Research Part E: Logistics and Transportation Review, Volume 43, Issue 6, 2007, Pages 687-709, ISSN 1366-5545, https://doi.org/10.1016/j.tre.2006.04.004.

[18] A. E. Jimenez-Cano, J. Martin, G. Heredia, A. Ollero, and R. Cano, "Control of an aerial robot with multi-link arm for assembly tasks," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2013.

[19] M. Bisgaard, A. la Cour-Harbo, and J. Dimon Bendtsen, "Adaptive control system for autonomous helicopter slung load operations," *Control Eng. Pract.*, vol. 18, no. 7, pp. 800–811, 2010.

[20] N. Michael, J. Fink, and V. Kumar, "Cooperative manipulation and transportation with aerial robots," *Auton. Robots*, no. September 2010, pp. 1–14, 2010.

[21] I. Palunko, R. Fierro, and P. Cruz, "Trajectory generation for swing-free maneuvers of a quadrotor with suspended payload: A dynamic programming approach," *Proc. - IEEE Int. Conf. Robot. Autom.*, pp. 2691–2697, 2012.

[22] B. S. Rego and G. V. Raffo, "Suspended Load Path Tracking by a Tilt-rotor UAV"., *IFAC-PapersOnLine*, vol. 49, no. 32, pp. 234–239, 2016.

[23] D. Mellinger, M. Shomin, N. Michael, and V. Kumar, "Cooperative Grasping and Transport using Multiple Quadrotors-mellingerDARS10.pdf," pp. 1–14, 2015.

[24] P. E. I. Pounds, D. R. Bersak, and A. M. Dollar, "Grasping from the air: Hovering capture and load stability," *Proc. - IEEE Int. Conf. Robot. Autom.*, pp. 2491–2498, 2011.

[25] Q. Lindsey, D. Mellinger, and V. Kumar, "Construction of Cubic Structures with Quadrotor Teams," *Robot. Sci. Syst. VII*, 2011.

[26] M. Zhao, K. Kawasaki, X. Chen, S. Noda, K. Okada, M. Inaba, and A. T. M. T. Multi-, "Whole-body Aerial Manipulation by Transformable Multirotor with Two-dimensional Multilinks" *Proc. - IEEE Int. Conf. Robot. Autom.*, pp. 5175–5182, 2017.

[27] M. Zhao, K. Kawasaki, K. Okada, and M. Inaba, "Transformable multirotor with two-dimensional multilinks : modeling , control , and motion planning for aerial transformation," *Adv. Robot.*, vol. 1864, no. July, p. 817, 2017.

[28] L. Suphachart, S. Shimahara, R. Ladig, and K. Shimonomura, "Vision Based Autonomous Orientational Control for Aerial Manipulation via On-board FPGA," *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2016.

[29] J. Thomas, J. Polin, K. Sreenath, and V. Kumar, "Avian-Inspired Grasping for Quadrotor Micro UAVs," *Vol. 6A 37th Mech. Robot. Conf.*, p. V06AT07A014, 2013.

[30] K. Kondak, K. Krieger, A. Albu-Schaeffer, M. Schwarzbach, M. Laiacker, I. Maza, A. Rodriguez-Castano, and A. Ollero, "Closed-loop behavior of an autonomous helicopter equipped with a robotic arm for aerial manipulation tasks," *Int. J. Adv. Robot. Syst.*, vol. 10, 2013.

[31] D. Lunni, A. Santamaria-Navarro, R. Rossi, P. Rocco, L. Bascetta, and J. Andrade-Cetto, "Nonlinear model predictive control for aerial manipulation," *2017 International Conference on Unmanned Aircraft Systems (ICUA*S), 2017.

[32] S. Kim, H. Seo, S. Choi, and H. J. Kim, "Vision-Guided Aerial Manipulation Using a Multirotor With a Robotic Arm," *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 4, pp. 1912–1923, 2016.

[33] C. Korpela, P. Brahmbhatt, M. Orsag, and P. Oh, "Towards the realization of mobile manipulating unmanned aerial vehicles (MM-UAV): Peg-in-hole insertion tasks," *IEEE Conf. Technol. Pract. Robot Appl. TePRA*, pp. 0–5, 2013.

[34] F. Huber, K. Kondak, K. Krieger, D. Sommer, M. Schwarzbach, M. Laiacker, I. Kossyk, S. Parusel, S. Haddadin, and A. Albu-Schaffer, "First analysis and experiments in aerial

manipulation using fully actuated redundant robot arm," *IEEE Int. Conf. Intell. Robot. Syst.*, pp. 3452–3457, 2013.

[35] V. Lippiello and F. Ruggiero, "Exploiting redundancy in Cartesian impedance control of UAVs equipped with a robotic arm," *IEEE Int. Conf. Intell. Robot. Syst.*, pp. 3768–3773, 2012.

[36] S. Kim, S. Choi, and H. J. Kim, "Aerial manipulation using a quadrotor with a two DOF robotic arm," *IEEE Int. Conf. Intell. Robot. Syst.*, pp. 4990–4995, 2013.

[37] G. Heredia, A. Jimenez-Cano, I. Sanchez, D. Llorente, V. Vega, J. Braga, J. Acosta, and A. Ollero, "Control of a multirotor outdoor aerial manipulator," *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014.

[38] M. Orsag, C. M. Korpela, S. Bogdan, and P. Y. Oh, "Hybrid adaptive control for aerial manipulation," *J. Intell. Robot. Syst. Theory Appl.*, vol. 73, no. 1–4, pp. 693–707, 2014.

[39] R.Mebarki and V.Lippiello, "Image-based control for aerial manipulation," *Asian J. Control*, vol. 16, no. 3, pp. 646–656, 2014.

[40] V. Lippiello, J. Cacace, A. Santamaria-Navarro, J. Andrade-Cetto, M. Á. Trujillo, Y. R. Esteves, and A. Viguria, "Hybrid Visual Servoing With Hierarchical Task Composition for Aerial Manipulation," *IEEE Robot. Autom. Lett.*, vol. 1, no. 1, pp. 259–266, 2016.

[41] M. Laiacker, F. Huber, and K. Kondak, "High accuracy visual servoing for aerial manipulation using a 7 degrees of freedom industrial manipulator," *IEEE Int. Conf. Intell. Robot. Syst.*, vol. 2016–Novem, pp. 1631–1636, 2016.

[42] A. Santamaria-navarro, P. Grosch, V. Lippiello, J. Solà, and J. Andrade-cetto, "Uncalibrated Visual Servo for Unmanned Aerial Manipulation," vol. 4435, no. c, pp. 1–11, 2017.

[43] G. Darivianakis, K. Alexis, M. Burri, and R. Siegwart, "Hybrid predictive control for aerial robotic physical interaction towards inspection operations," *Proc. - IEEE Int. Conf. Robot. Autom.*, pp. 53–58, 2014.

[44] C. Papachristos, K. Alexis, and A. Tzes, "Technical Activities Execution with a TiltRotor UAS employing Explicit Model Predictive Control," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 11036–11042, 2014.

[45] C. Papachristos, K. Alexis, and A. Tzes, "Efficient force exertion for aerial robotic manipulation: Exploiting the thrust-vectoring authority of a tri-tiltrotor UAV," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2014.

[46] C. M. Korpela, T. W. Danko, and P. Y. Oh, "MM-UAV: Mobile manipulating unmanned aerial vehicle," *J. Intell. Robot. Syst. Theory Appl.*, vol. 65, no. 1–4, pp. 93–101, 2012.

[47] P. O. Pereira, R. Zanella, and D. V. Dimarogonas, "Decoupled design of controllers for aerial manipulation with quadrotors," *IEEE Int. Conf. Intell. Robot. Syst.*, vol. 2016–Novem, no. 644128, pp. 4849–4855, 2016.

[48] M. Kamel, S. Comari, and R. Siegwart, "Full-body multi-objective controller for aerial manipulation," in *24th Mediterranean Conference on Control and Automation, MED 2016*, 2016, pp. 659–664.

[49] Davies, K. (2017). *Performance Analysis of Nonlinear Model Predictive Control Applied to Multi-Rotor Unmanned Aerial Vehicles*. Ph.D. University of Calgary.

[50] A. Kelly, *Mobile robotics: mathematics, models, and methods*. New York, NY: Cambridge Univ. Press, 2013.

[51] Zeinali, M., & Notash, L. (2010). Adaptive sliding mode control with uncertainty estimator for robot manipulators. *Mechanism and Machine Theory*, *45*(1), 80–90.

[52] *Virtual Robot Experimentation Platform USER MANUAL*, Version 3.4.0., Coppelia Robotics GmbH., Zurich., Switzerland, 2017.

[53] Moosavian, S. A. a., & Papadopoulos, E. (2007). Free-flying robots in space: an overview of dynamics modeling, planning and control. *Robotica*, *25*(5), 537–547. https://doi.org/10.1017/S0263574707003438