

8-2014

Unified Dynamics and Control of a Robot Manipulator Mounted on a VTOL Aircraft Platform

Peng Xu

Clemson University, pxu@g.clemson.edu

Follow this and additional works at: https://tigerprints.clemson.edu/all_dissertations



Part of the [Electrical and Computer Engineering Commons](#)

Recommended Citation

Xu, Peng, "Unified Dynamics and Control of a Robot Manipulator Mounted on a VTOL Aircraft Platform" (2014). *All Dissertations*. 1321.

https://tigerprints.clemson.edu/all_dissertations/1321

This Dissertation is brought to you for free and open access by the Dissertations at TigerPrints. It has been accepted for inclusion in All Dissertations by an authorized administrator of TigerPrints. For more information, please contact kokeefe@clemson.edu.

UNIFIED DYNAMICS AND CONTROL OF A ROBOT MANIPULATOR MOUNTED ON A VTOL AIRCRAFT PLATFORM

A Dissertation
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
Electrical Engineering

by
Peng Xu
August 2014

Accepted by:
Dr. Timothy C. Burg, Committee Chair
Dr. Richard E. Groff, Co-Chair
Dr. Ian D. Walker
Dr. John R. Wagner

Abstract

An innovative type of mobile manipulator, designated Manipulator on VTOL (Vertical Take-Off and Landing) Aircraft (MOVA), is proposed as a potential candidate for autonomous execution of field work in less-structured indoor and outdoor environments. Practical use of the MOVA system requires a unified controller that addresses the coupled and complex dynamics of the composite system; especially the interaction of the robotic manipulator with the aircraft airframe. Model-based controller design methods require explicit dynamics models of the MOVA system.

Preliminary investigation of a two-dimensional MOVA system toward a dynamics model and controller design is presented in preparation for developing the controller of the more complex MOVA system in 3D space. Dynamics of the planar MOVA system are derived using the Lagrangian approach and then transforming the result into a form that facilitates controller design using the concept of a virtual manipulator. A MOVA end-effector trajectory tracking controller was designed with the transformed dynamics equation using the integrator back-stepping control design framework. Validity of the controller is shown via stability analysis, simulation results, and results from a physical test-bed.

A systematic approach is illustrated for the derivation of the 3D MOVA system dynamics equations. The resulting dynamics equations are represented abstractly in the standard robot dynamics form and proven to have the skew-symmetric property,

which is a useful property for control derivation. An open source Mathematica program was developed to achieve automatic symbolic derivation of the MOVA system dynamics. Accessory tools were also designed to create a tool-chain that starts with an Autodesk Inventor CAD drawing, generates input to the Mathematica program, and then formats the output for direct use in MATLAB and Simulink. A unified nonlinear control algorithm that controls the 3D MOVA system, including both the aircraft and the onboard manipulator, as a single entity was developed to achieve trajectory tracking of the MOVA end-effector position and attitude based on the explicit dynamics equation. Globally Uniformly Ultimately Bounded (GUUB) stability is proven for the controller using Lyapunov-type stability analysis. Physical testing was constructed in order to demonstrate the performance of the proposed controller on a MOVA system with a two-link onboard manipulator.

Dedication

This thesis is dedicated to my parents, who provide me with their best, give me reasonable freedom through my growth, and encourage and support me during my hard times.

Acknowledgments

I would like to express my sincere gratitude to my co-advisors Dr. Timothy Burg and Dr. Richard Groff for providing me with the opportunity of participating in this research and their motivation and patience throughout this project. Their encouragement and guidance were always available especially at times when difficulties and confusion appeared during the research project and the process of writing this dissertation. I would like to thank my committee members Dr. Ian Walker and Dr. John Wagner for their encouragement and insightful comments.

I would like to thank my partner in this project, Ran Huang, for a great job in design, construction and experimentation of the physical system test-bed. I would also like to give my thanks to those who worked together in the lab for the stimulating discussions and encouragement through their interests in this project.

Lastly, and most importantly, I wish to thank my parents for their support and love.

Table of Contents

Title Page	i
Abstract	ii
Dedication	iv
Acknowledgments	v
List of Tables	viii
List of Figures	ix
1 Introduction	1
1.1 Motivation	1
1.2 System Description	4
1.3 Previous Work	10
1.4 Organization	16
2 Dynamics and Control of Planar MOVA System	18
2.1 Introduction	18
2.2 Kinematics of the Planar MOVA	21
2.3 Dynamics of the Planar MOVA	31
2.4 Controller Design for the Planar MOVA	42
2.5 Experimental Results	53
2.6 Conclusion	62
3 Systematic Approach to MOVA System Dynamics Derivation	64
3.1 Introduction	64
3.2 System Description and Kinematics	67
3.3 Derivation of Dynamics Equation	77
3.4 Automation of Dynamics Derivation	87
3.5 Validation of Derived Dynamics Equation	96
3.6 Conclusion	108

4	Trajectory Tracking Control of the MOVA System	110
4.1	Introduction	110
4.2	Controller Design	114
4.3	Stability Analysis	128
4.4	Simulation	131
4.5	Conclusion	144
5	Conclusions	145
5.1	Future Work	147
	Appendices	149
A	Example Usage of MOVADYN and Its Auxiliary Programs	150
	Bibliography	155

List of Tables

2.1	Notation description	23
2.2	Simulation Plant Physical Parameters	38
2.3	Physical Parameters of Planar MOVA test-bed	59
3.1	List of notation	70
3.2	Names of matrices and vectors in MOVADYN output	94
3.3	Physical Parameters of the 3D MOVA system used for validation. . .	104

List of Figures

1.1	Number of work-related fatal injury events caused by falling. Plotted from data published by U.S. Bureau of Labor Statistics[1].	2
1.2	Illustration of several potential applications of the MOVA system. A) MOVA changing light bulb on a road lamp; B) MOVA installing new antenna onto the mobile signal tower; C) MOVA defusing a bomb located on top of a vending machine in a subway station, an inaccessible location for ground-based mobile manipulators; D) MOVA flying into a factory through a window and shutting down a malfunctioning machine.	5
1.3	Body-fixed thrust force and torques abstraction of varies types of VTOL aircraft.	6
1.4	MOVA system with a VTOL and two DOF onboard manipulator is equivalent to 6 DOF manipulator with unlimited workspace.	8
2.1	Coordinate system definition for the planar MOVA system. \mathcal{I} indicates the earth fixed coordinate frame and \mathcal{B} represents body-fixed coordinate frame.	22
2.2	Real manipulator kinematics chain versus virtual manipulator kinematics chain for a general N-link MOVA system. Both chains reach the same endpoint with the same orientation.	26
2.3	View of a single link of the virtual manipulator and original manipulator. The corresponding links never rotate relative to each other.	28
2.4	Block diagram of dynamics validation test-bed.	37
2.5	Plot of dynamics validation result with random input signals.	39
2.6	Plot of dynamics validation result with the sine wave test input signal.	40
2.7	Plot of p_e with the sine wave test input signal.	40
2.8	Plot of dynamics validation result with the constant test input signal.	41
2.9	Plot of p_e with the constant test input signal.	42
2.10	Trajectory of end-effector for simulation is shown as the dashed circle. The tip of arrow (not to scale) denotes the end-effector of the MOVA system.	55
2.11	End-effector position error projected onto x- and z-axis of \mathcal{I}	56
2.12	Norm of end-effector position error	56
2.13	Plot of $\theta_{01r}(t)$, the reference trajectory of end-effector orientation, $\theta_{01}(t)$, the actual trajectory and $e_{01}(t)$, the tracking error.	57

2.14	Illustration of the proposed hardware-in-the-loop test bed. The light colored cylinders represent passive joints of the test-bed, which also serve in measurement of VTOL position and orientation. The triple lines represents links of the test-bed while single lines illustrate frames and the manipulator link of the 2D MOVA system. The dark colored cylinder is the joint of the onboard manipulator.	58
2.15	Photo of the planar MOVA hardware-in-the-loop test-bed.	59
2.16	Comparison of norm of end-effector position error at steady state from $t = 25$ sec to $t = 45$ sec	61
2.17	Comparison of VTOL aircraft coordinate x_0 . The reference trajectory of end-effector attitude is added to show the direct effect of dynamics disturbance of the onboard manipulator on the VTOL aircraft.	62
3.1	Illustration of a general MOVA system frame definition and notations conventions	69
3.2	Flow chart of MOVADYN and its auxiliary program	89
3.3	Illustration of MOVADYN code generation facility	92
3.4	Block diagram of dynamics validation test-bed.	102
3.5	Simulink diagram of the SimMechanics simulation.	103
3.6	Illustration of a 3D MOVA system with 2DOF onboard manipulator.	104
3.7	Plot of dynamics validation result with random input signals.	105
3.8	Plot of dynamics validation result with the sine wave test input signal.	106
3.9	Plot of dynamics validation result with the constant test input signal.	107
3.10	Plot of end-effector position p_E from MOVADYN derived model with the constant test input signal.	108
4.1	Screenshot of the VRML viewer during simulation.	133
4.2	Plot of of position error of end-effector norm.	135
4.3	Plot of orientation error of end-effector in terms of minimum rotation angle.	135
4.4	Spatial visualization of actual and reference trajectory of both end-effector and CM of entire MOVA system.	136
4.5	Visualization of the composite position and orientation trajectory. The thick blue curve is the position trajectory of the end-effector. The orientation trajectory is marked out sparsely with small frame icons attached to the position trajectory. The blue line the is z -axis, and green line is the y -axis. the z -axis is always directed to the fixed point	138
4.6	Plot of position error of end-effector in term of distance.	139
4.7	Plot of orientation error of end-effector in term of minimum rotation angle.	140
4.8	Visualization of the actual and reference trajectory of end-effector position in 3D space.	141
4.9	Plot of position error of end-effector in term of distance.	142

4.10	Plot of orientation error of end-effector in term of minimum rotation angle.	143
A.1	3D model of a MOVA system with a RRR onboard manipulator. . . .	152

Chapter 1

Introduction

An innovative type of aerial vehicle system, designated Manipulator on VTOL (Vertical Take-Off and Landing) Aircraft (MOVA), is proposed as a potential candidate for autonomous execution of field work in less-structured indoor and outdoor environments. To maximize load capacity of the MOVA system, a unified and minimized design is sketched. Model-based approach is adopted for design of a controller for the complex and coupled system dynamics, which requires that a dynamics model of the MOVA system be derived. A brief reading guide is provided at the end of this chapter.

1.1 Motivation

Data from U.S. Bureau of Labor Statistics between 1992 and 2010 (plotted in Fig. 1.1) [1] suggests more than six hundred fatal work-related injuries occur every year due to falling, which ranks as the third most frequent cause of fatal work injury. Tasks related to maintenance, construction, and inspection, for example, changing lights or components on a tower, expose a worker to height hazards. The field of

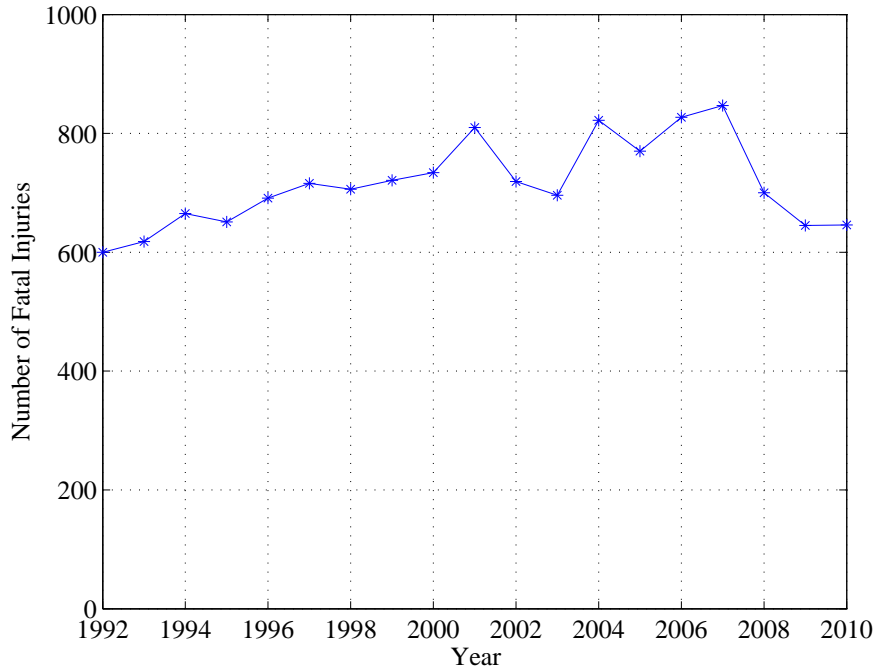


Figure 1.1: Number of work-related fatal injury events caused by falling. Plotted from data published by U.S. Bureau of Labor Statistics[1].

robotics has grown to replace humans in such dangerous environments. For example, small VTOL aircraft were recently exploited for inspection after the accident at the Fukushima Nuclear Power Plant in Japan caused by the earthquake and the induced tsunami.

Advances in robot manipulators, sometimes called robot arms, have propelled manufacturing forward in gigantic steps since the introduction of the first industrial models in the 1950s. Equipped with superior accuracy, reliability, and endurance, compared to a person, they have been widely deployed in factory assembly lines and other industrial settings to perform repetitive, precision, or dangerous tasks. Technology improvements in software and hardware have spawned new applications in scenarios that have similar requests for automation, precision and reliability, such as autonomous package handling in warehouses and surgical procedures in hospital oper-

ating rooms. However, traditional fixed-base installations impose a severe workspace limitation proportional to the size of the manipulator. In addition, most manipulators are designed to work in highly structured environments and demand setup and calibration procedures when deployed into new locations. Thus, conventional robot manipulators have small utility in field work scenarios, in which covering large workspaces and the ability to adapt to environmental changes are generally required.

Mobile robots are built and programmed to move about the environment and have theoretically infinite workspace. Mobile manipulators, a fusion of a robot manipulator and a mobile robot platform, suggest the promise of moving the accuracy and reliability into the field by extending the limited workspace of the manipulator. Selection of the mobile robot platform and the manipulator is largely application specific.

VTOL aircraft platforms have merits over the ground-based ones, because as an aerial robot, it is able to travel in three-dimensional space which extends the workspace above ground-based mobile robots to arbitrary height levels. Use of an aerial vehicle minimizes terrain related issues and can approach a target location from the shortest path, which is a big advantage in terms of viability and agility.

There appears to be a specific opportunity to combine a VTOL platform with a manipulator base, this system will be referred as manipulator on VTOL aircraft (MOVA). The unique features of the VTOL would enable the hybrid MOVA system to address off-the-ground scenarios. Besides the ability of performing tasks of a ground vehicle based mobile manipulator, the MOVA is especially adaptable to tasks that take place where no negotiable terrain path leads to the work site. Examples include the top of a water tower, half-way up a cliff, in a locked building with open windows, and inside of a cave with rough ground. Thus, the MOVA system would be able to carry out many tasks that are not feasible otherwise. For instance, a MOVA system

could be adopted to replace staff who work on power transmission line towers, radio signal towers, roof-top of buildings, scaffolds or other aloft positions. A MOVA system could potentially physically interact with the environment and assist the recovery of plant function. The MOVA could also serve in exploration or object manipulation tasks in less human-friendly environments, such as bomb searching and defusing (see Fig 1.2).

1.2 System Description

The proposed MOVA system is a new type of mobile manipulator and there are few existing implementations or guidelines for how the system should be structured. Similar applications include a manipulator on an underwater vehicle and a manipulator on a satellite. The underwater system is significantly different from the MOVA in the amount of damping from the water and the potential for neutral buoyancy of the system. The space systems usually have over-actuated satellite bases and manipulators with at least six degrees-of-freedom and are deployed in a gravity free environment. The main concern is to avoid application of attitude correction mechanism on satellite by careful path planning that reduces the effects of dynamic reaction of the satellite manipulator. In this section, a unified, minimized form of the MOVA system is sketched in order to ensure the highest load capacity of the mobile manipulator. Then the control problem of such MOVA system naturally arises as the resulting MOVA system has a complex dynamics structure and is not suitable to be controlled by independent arm and platform controllers. The resulting MOVA dynamics are complex due to the interaction between the manipulator and the moving aircraft base, that is, the force resulting from motion of the manipulator arm are large relative to the inertia of the VTOL platform. The complex dynamics interaction will

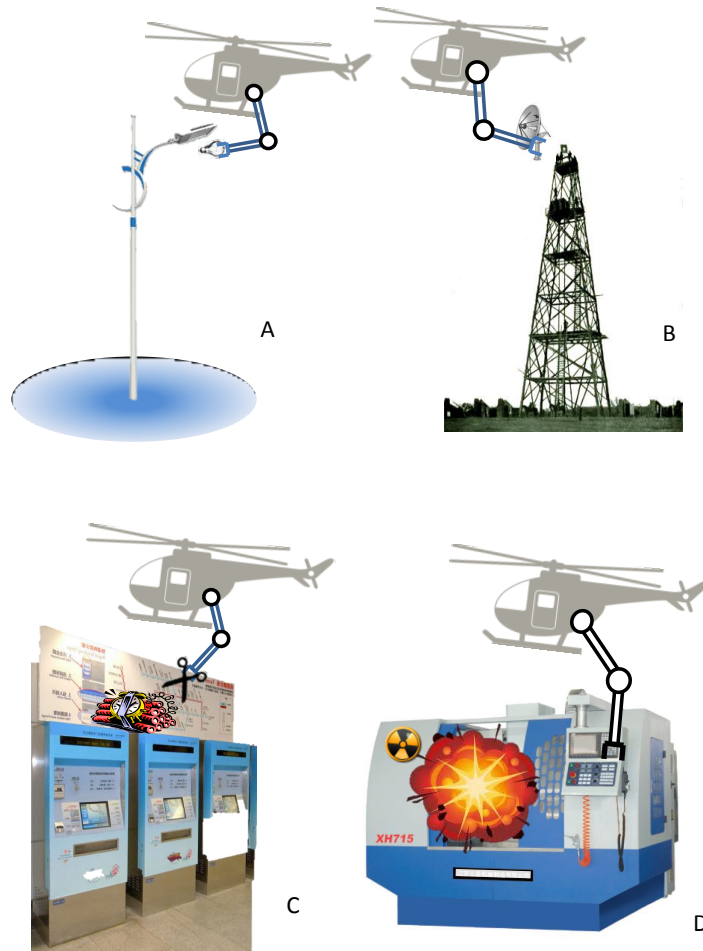


Figure 1.2: Illustration of several potential applications of the MOVA system. A) MOVA changing light bulb on a road lamp; B) MOVA installing new antenna onto the mobile signal tower; C) MOVA defusing a bomb located on top of a vending machine in a subway station, an inaccessible location for ground-based mobile manipulators; D) MOVA flying into a factory through a window and shutting down a malfunctioning machine.

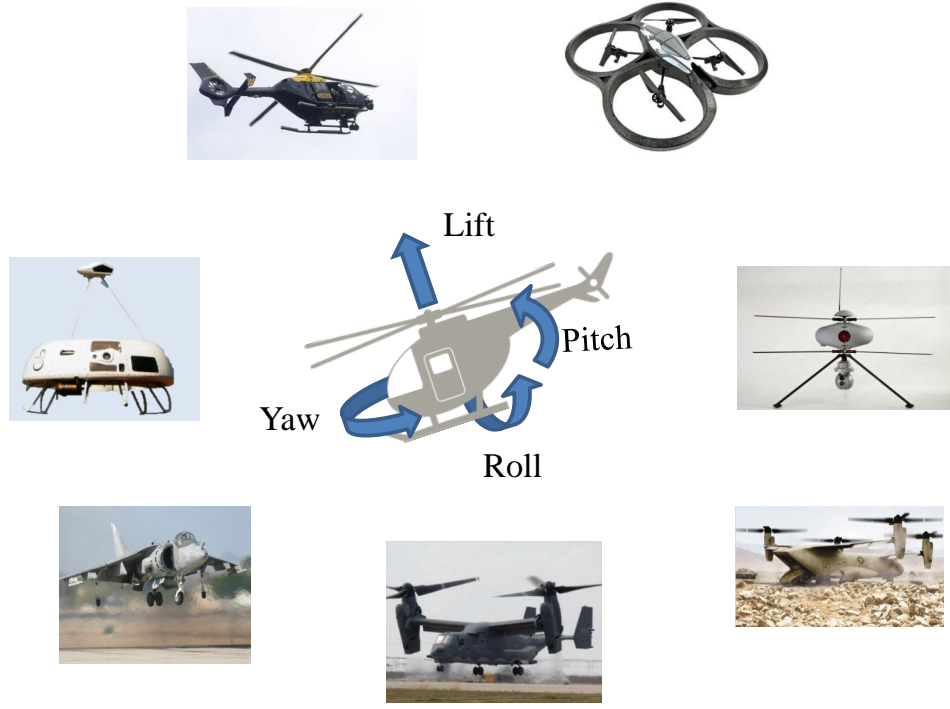


Figure 1.3: Body-fixed thrust force and torques abstraction of various types of VTOL aircraft.

likely prohibit MOVA from being directly controlled by operator, but will also pose an interesting control problem with unique characteristics. In order to generalize the discussion, various types of VTOL aircraft are abstracted by one free-floating body with a body-fixed thrust vector and torques direction of three orthogonal axis as shown in Fig. 1.3.

The load capacity of the MOVA system, the ability of the MOVA system to manipulate objects by its end-effector, is determined primarily by payload capacity of VTOL aircraft and the design of onboard manipulator. In order to maximize load capacity of MOVA system specification, two major qualitative criterion are considered: the ratio between maximum VTOL aircraft and MOVA system self-weight should be high to allow for heavier payload and necessary maneuver room of VTOL aircraft;

and the onboard manipulator ought to be designed so that with certain motor ratings the end-effector is able to have maximum force or torque for payload manipulation.

Increase of lift and weight ratio can be achieved by adopting more efficient means of lift force generation and more importantly reducing the total system mass. Higher lift force quite often leads to more weight for the power system, which includes the fuel (or battery), engine (or motor) and transmission and represent trade-offs that must be balanced in any design. On the other hand, the efficiency of the thruster is limited by available technology and is not easily improved in the design of a specific aircraft. The most feasible path to improve lift to weight ratio is to reduce weight of components. Minimizing system total weight can be done by applying lighter materials for construction, designing better structure and mechanism.

One important method for reducing total system weight, that also helps meet the payload capability requirements of the manipulator, is to reduce the number of joints of the onboard manipulator and “borrow” the lost degrees-of-freedom from the VTOL aircraft that carries the manipulator. Less joints will lead to fewer motors and transmission parts for the manipulator, a nontrivial reduction in weight for the whole system. Moreover, since a joint closer to the base also has to provide force or torque that drives the joints downstream to the end-effector, a smaller number of joints is more efficient in the sense that available torque and force are better transferred to the payload at the end-effector.

To minimize the weight, construction of a minimized MOVA system designed is proposed. The attached manipulator will have the minimum necessary number of joints without losing ability for the MOVA end-effector to move to any point in three-dimensional space at any orientation, i.e., still provide 6DOF movement of the end-effector. At the extreme, is a manipulator with no joints at all, i.e. the end-effector is directly attached to VTOL aircraft, where the work of placing end-

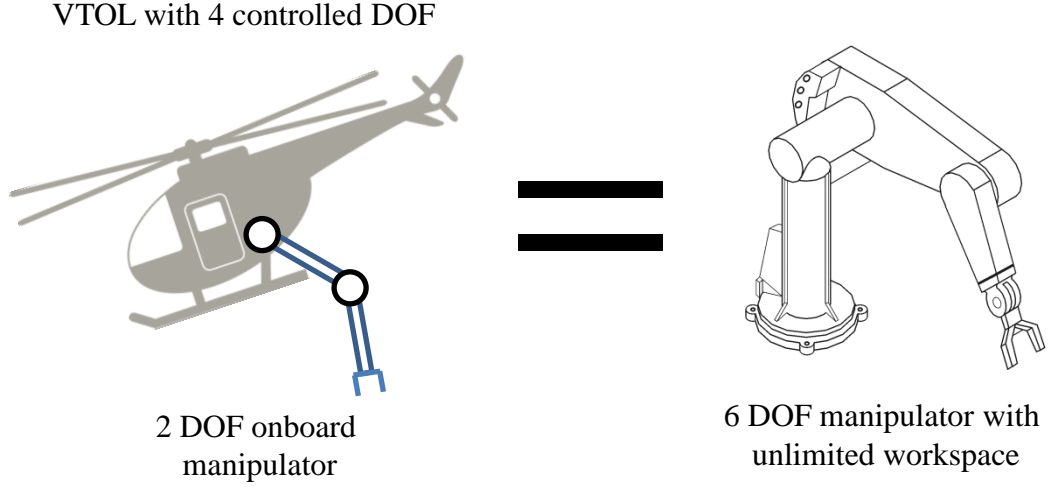


Figure 1.4: MOVA system with a VTOL and two DOF onboard manipulator is equivalent to 6 DOF manipulator with unlimited workspace.

effector falls entirely to the VTOL aircraft. However, this is not achievable due to the capabilities of the VTOL aircraft. Though the VTOL aircraft is a free-body in three-dimensional space with six degree-of-freedom, two of these, the pitch and roll angles, cannot be independently controlled. Only three translational DOF and yaw movement can be independently controlled for the aircraft types shown in Fig. 1.3. The pitch and roll of the VTOL aircraft are dedicated to translation movement of the aircraft via reorientation of the body-fixed thrust vector. Consequently, the aircraft body is only able to provide four controlled degree-of-freedom to the end-effector of manipulator. In other words, two additional DOFs are expected for the end-effector to track an arbitrary position and orientation trajectory in three-dimensional space. This fact dictates that the minimum number of DOFs for the attached manipulator to, required for the MOVA to perform the trajectory tracking objective, is two revolute joints. This concept of the MOVA as a 6DOF manipulator with unlimited workspace is depicted in Fig. 1.4.

For practical purposes, the MOVA system has to be automatically controlled, or at least assisted, since complexity of the coupled multi-DOF dynamics of the MOVA will prohibit it from being effectively controlled directly by human operators. A fully autonomous MOVA is desirable for increased cost and efficiency. Motion control of MOVA focuses on establishing end-effector position and orientation, a common yet important precursor to establishing higher levels of task automation, which will be specific to individual applications.

A separate control strategy is adopted in many existing mobile manipulators applications, especially those built on ground or under-water vehicles. This approach is equivalent to controlling onboard manipulator as a fixed-base robot while the base is moving and simultaneously controlling the carrying vehicle. The dynamical disturbance from the onboard manipulator unmodeled, which is only justified if the carrying vehicle has significant more mass than the manipulator or is kinematically constrained to an object that has such property so that the disturbance from manipulator is negligible for the carrying vehicle. Rather than a strategy of controlling the mobile platform and manipulator separately and regarding movement of each others as disturbances on the other, an integrated controller that coordinates control of the VTOL aircraft and the manipulator is necessary.

In sum, the MOVA system proposed, which joins advantages from VTOL aircraft and robot manipulators, will be suitable for many field tasks that have not been achieved with help of robots. These tasks poses fatal hazards to human workers and cannot be established using current robotics technology. A sketched design of the MOVA is made which contains onboard manipulator with minimal number of links yet is still able to achieve end-effector trajectory tracking. Application of MOVA system requires unified model-based controller due to coupled and complex dynamics. Model-based controller design methods also requests the dynamics model of the MOVA

system to be investigated.

1.3 Previous Work

The need for a MOVA system is compelling and the opportunity is promising. Study of the dynamics and control strategy for robot manipulator or similar mechanisms mounted on VTOL aircraft became an emerging field after years of extensive research on unmanned aerial vehicle (UAV) systems [2, 3, 4, 5]. Some recent literature shows the possibility of such systems while also illustrating the importance of appropriate handling of dynamics interaction between the onboard manipulator and the aircraft, and variation of system inertia with change of manipulator configuration. Lee *et al.* has introduced the concept of coordinated control for a UAVARM system and provided theoretical work on controller design [6]. Korpela and his colleagues performed experiments of a system composed of a quadrotor and two 4DOF onboard manipulators to achieve “grab and drop” of object [7]. Preliminary dynamics analysis and results from a PI controller with additional velocity inner-loop and feedforward terms are presented. Later, the same group experimented with a larger scale model helicopter and a 7DOF redundant manipulator using a similar control scheme [8]. Results from both experiments suggest a model-based controller developed with more accurate dynamics model of the system will improve the control performance. Thomas *et al.* proposed a specialized biomimetic system composed of quadrotor and a one-link robot manipulator with active claw-like gripper to perform object retrieval task when the entire system is in continuous flight [9]. Researchers Mellinger *et al.* and Pounds *et al.* separately investigated control problem of a system composed of a VTOL aircraft and a directly-attached gripper in an object pick-and-place task scenario [10, 11]. Both of them focused on resolving the issue related to varying system inertia in run-

time, with Mellinger proposed an online estimation system and Pounds worked on theoretical analysis of system stability bounds under disturbance of payload. Simplified dynamics models are used in all the above mentioned research. Either planar system dynamics model of a 3D system are adopted or dynamics interaction between manipulator and the carrier VTOL aircraft is only partially considered. Conceded that it is justifiable to use a simplified model for controller development and consider the unmodeled dynamics as disturbances to the system, using more accurate models will likely to improve system performance. A full-blown 3D dynamics model of the MOVA system is necessary in construction simulation as well.

On the dynamics modeling side, the MOVA system can be categorized as a multi-body system, the dynamics of which have been extensively studied in the past. Some specific multi-body systems, such as satellite-based manipulators, and underwater vehicle-based manipulators, have strong similarities to the MOVA system and share similar structures, and thus provide an initial point of reference.

The Euler-Lagrangian approach based on Hamilton's principle is a well-known general approach for deriving multi-body system dynamics [12, 13]. Given a set of independent general coordinates describing system configuration and the Lagrangian expression, which for a mechanical system is the difference between kinetic energy and potential energy, the dynamics of each coordinate variable can be systematically evaluated. The benefit of the Euler-Lagrangian approach over the Newtonian method is that interactions among bodies are intrinsically handled and do not need special attention. The difficulty in directly applying the Euler-Lagrangian resides in finding a description of the kinetic and potential energy by a set of suitable coordinates and handling the larger number of terms this approach yields.

The MOVA system is postulated as a serial manipulator connected to a VTOL aircraft. For serial fixed-based robotic manipulators, kinematics is often abstracted

by a set of Denavit-Hartenberg (D-H) parameters for concise representation. These parameters can be substituted into a set of general form equations from which the kinematics and differential kinematics of the manipulator can be derived. The kinematics and differential kinematics are then used for systematic derivation of the manipulator dynamics equation [14]. It is well known that the dynamics of a system composed of multiple bodies cannot be formed by trivially stacking together dynamics of single components. Interaction between constraint bodies can create new terms in the dynamics equation that do not exist in dynamics of individual bodies. Thus, the dynamics of MOVA cannot be found by simply combining the individual dynamics of onboard manipulator and the VTOL aircraft, which is usually modeled as rigid-body with six degree-of-freedom.

Researchers of other mobile manipulator systems have faced the same challenge and provided hints for a solution. Space-based manipulator communities were the first to investigate mobile manipulator dynamics by augmenting the general dynamics derivation with a framework that helps with description of the Lagrangian energy equation and yields clearer result in a more compact format via hierarchical representation [15]. One core idea of this framework is representation of the translational displacement of any points in the multi-body system with a virtual kinematic chain starting from an imaginary point called the virtual ground, which is the center of mass of the entire system. The kinematic chain is systematically derived with the barycenter representation. In contrast to forming the kinematics representation from a significant point on the mobile platform, such as the VTOL center of gravity, this approach yields simpler dynamic equations. In this methodology, the translational dynamics of the center of mass and the rotational dynamics of a body-frame affixed to the center of mass is decoupled from rest of the system naturally since internal forces and torques among bodies do not have an effect on the center of mass. It is

desirable to have dynamics equations in a less coupled form to facilitate controller development. Moreover, a special kinematic chain which starts from the virtual ground and ends at the end-effector of the manipulator is called the virtual manipulator [16]. The virtual manipulator includes one more spherical joint than the actual onboard manipulator. The spherical joints represents rotation of the mobile vehicle on which the manipulator is attached. With the spherical joint rotate according to the mobile vehicle attitude and the other joints having the same joint parameters as the actual manipulator, the virtual manipulator has the same end-effector position and orientation as the actual manipulator. Aided by the systematic approach and compact representation, computer program was developed based on this framework to aid in the derivation of system dynamics equations for space manipulators in analytical form [17].

Researchers of Unmanned Under-water Vehicle (UUV) based manipulator have been utilizing the similar approach for dynamics modeling, yet concerns more about the dynamics involving interaction with the surrounding liquid [18, 19, 20]. The relatively high density of water inevitably induces hydrodynamics terms that cannot be ignored.

Besides deriving dynamics model analytically, numerical method also exist for system simulation purposes. Given physical parameters of individual parts of multi-body system and set of constraints, evolution of the dynamical system can be calculated [21]. This numerical method does not output closed form dynamics model, which will be needed for dynamics property analysis and controller design, though it offers ability to simulate complex multi-body system as long as computational power permits. The result of simulation may be also utilized for validation of analytically derived model when direct observation of system is not yet available.

Similarity in system structure and method for dynamics modeling may suggest

that controller design of the MOVA system can be found in literature on space and underwater mobile manipulators. However, this has been found not true after survey of the research documents. The control objective for MOVA is quite different from that of manipulators deployed on satellites and space stations. Much of the space manipulator control research are focuses on minimizing use of the attitude control fuels, as they are precious and the major limiting factor of satellite system life-span. Enhanced Disturbance Map (EDM) was developed by Dubowsky and Torres to enable a heuristic path planning approach for minimization of dynamic disturbance of manipulator on the attached space vehicle [22]. Zero Reaction Maneuver (ZRM) was proposed in order to generate trajectories for the space manipulator that has zero reaction on the carrying vehicle [23]. However, this technique turns out to have limited application because ZRM trajectory for 6 DOF manipulator only exists in special cases. Besides generating an optimal trajectory to minimize the reaction from the onboard manipulator on the space vehicle, a dynamic balance control strategy was presented by Huang *et al.* which utilizes additional manipulator to counter-react the dynamic disturbance. One onboard manipulator is used as the working manipulator at a time while the other move accordingly to generates exact opposite disturbance to the space vehicle. Attitude control fuel can be saved since the manipulator is powered by electricity which can be regenerated via solar panel on the satellite. A summary of planning and controlling of free-flying and free-floating space robotic system was presented with examples in [24]. Literature about control of underwater vehicle based manipulator mainly employ separated control strategy for the vehicle and manipulator. Topics of control for such system was initially investigated in [18], in which feedback linearized controller were designed for vehicle and manipulator separately. In publication of Wilson *et.al.*, effects of coupling dynamics was quantitatively evaluated along with the proposed separate controller. Neglecting of coupling

dynamics in underwater manipulator control research may be due to the fact that the mass of underwater vehicle and the associated hydrodynamical added mass causes dominance of the underwater vehicle dynamics over that of the onboard manipulator. That is, the dynamic disturbance of manipulator on the underwater vehicle is small comparatively and thus can be ignored.

Although research of other mobile manipulator do not help control derivation of the MOVA, inspiration for control formulation of the MOVA system can be found in VTOL aircraft control research. Literature about VTOL aircraft control serves as an important resource because the MOVA system contains components of nonlinear, under-actuated, non-minimum-phase dynamics that originate from the VTOL aircraft. Many attempts have been made for VTOL aircraft control using different types of controllers. Erginer and colleagues published proportional-derivative controller applied to quadrotor VTOL [25]. Bouabdallah *et al.* showed simulation and experimental results of their OS4 quadrotor for performance comparison of traditional proportional-integral-derivative (PID) controller and the linear quadratic regulator (LQR) in [26]). A nonlinear tracking controller for VTOL aircraft is proposed in [27] by Setlur *et al.* and proved to have globally uniformly ultimately bounded (GUUB) tracking error with continuous trajectory input. In [28], Saeki *et al.* displayed a two-step linearization method for VTOL aircraft control, which is based on a linear high gain approximation of a back-stepping controller. Bouabdallah *et al.* also presented results for quadrotor control with back-stepping and sliding-mode controller [29] and showed that back-stepping controller yields a more smooth control input than that from the sliding-mode controller. Extensions of back-stepping techniques were demonstrated in literature to address concerns about application specifics. An adaptive approach is mixed with back-stepping control algorithm to compensate in the control design for model parameter uncertainty [5, 4]. Visual servo controllers were

developed in [30, 31] to relieve the requirement of velocity measurement in application that only has VTOL position measurement via vision-based sensors.

Moreover, on the topics of 3D attitude control, quaternion based derivation is frequently used to avoid singularity issues inherited in representation that uses sequential rotation angles, such as Euler angle. Work of Joshi *et al.* presented a quaternion based robust controller that achieves three-axis attitude stabilization of a rigid spacecraft [32]. This controller is suitable for large-angle maneuver of satellite for its singularity free derivation and proved with global asymptotic stability (GAS). Fragopoulos and Innocenti investigated the stability issues of quaternion based attitude controller in [33], in which they adopted a discontinuous Lyapunov function to obtain desired GAS stability for inherited discontinuous control law of three-dimensional attitude control. Kristiansen *et al.* also demonstrated an attitude controller using quaternion derivation for micro-satellite application in [34, 35]. It employed a back-stepping style derivation and obtained a controller in the similar form as [32] and achieved asymptotic stability.

1.4 Organization

The goal of this work is to implement trajectory tracking controller for the MOVA system end-effector as the infrastructure for future application level research. Our endeavor starts from a planar case study presented in Chapter 2 in preparation for the more complicated three dimensional case. In this preliminary investigation, dynamics equation of a MOVA system in two-dimensional space is derived and a controller is designed to enable trajectory tracking of the end-effector of the planar MOVA system. The results from dynamics derivation and controller design are tested to ensure the worthiness of extending the work into full three dimensional

MOVA system. Then, in Chapter 3 a systematic approach for dynamics derivation of the 3D MOVA system is described. This approach is also implemented into computer programs in order to alleviate the overwhelming complexity of deriving multi-body high-DOF dynamics equation of MOVA system by hand. Auxiliary tools are also developed to connect dynamics derivation steps to upstream and downstream of a system design process. The 3D MOVA end-effector trajectory controller is developed on the base of the analytically derived dynamics equation in Chapter 4. In additional simulation result, a physical test-bed is constructed with major help from Ran Huang based on the planar MOVA model to evaluate the controller in real world. Construction of the test-bed is briefly described and the result is shown at the end of Chapter 2 to demonstrate the effective of the controller and the advantage over the strategy of controlling VTOL aircraft and the onboard manipulator separately. In the end, the content of this work and the importance of this work for future study is summarized in Chapter 5 as a conclusion.

Chapter 2

Dynamics and Control of Planar MOVA System

2.1 Introduction

Preliminary investigation about a planar MOVA, which consists of a two dimensional VTOL aircraft and a single link manipulator, is performed in preparation for the three dimensional case.

2.1.1 Previous Work

The Euler-Lagrangian approach based on Hamilton's principle is a well-known general method for deriving multi-body system dynamics [12, 13]. However, depending on the choice of the general coordinates, the resulting equations appear in various forms for the same system, some of which may not be suitable for controller derivation due to the complex coupling terms. In [16], an alternative way of representing kinematics of a manipulator mounted on a mobile structure (satellite in their case) was presented and was named a virtual manipulator. The kinematic chain of a vir-

tual manipulator starts from the virtual ground, which is the center of mass of the entire system, ends at the end-effector of the manipulator via a series of virtual manipulator links, and preserves the joint angle between consecutive links. The dynamics of the virtual ground is not susceptible to internal force and torque between manipulator joints and thus is beneficial to later controller design. Extending this idea, a framework based on the hierarchical helps with description of the Lagrangian energy equation and yields a clearer and more compact result [15]. Researchers of Unmanned Under-water Vehicle (UUV) based manipulators have been utilizing a similar approach for dynamics modeling [18, 19, 20]. In that work, the major concern is the dynamic interaction with the surrounding liquid.

Inspiration for control formulation for the MOVA system can be found in VTOL aircraft control research. Bouabdallah *et al.* presented results for quadrotor control with back-stepping and sliding-mode controllers [29] and showed that back-stepping controller yields a more smooth control input than that from the sliding-mode controller. Extensions of back-stepping techniques were demonstrated in literature to address concerns about application specifics. An adaptive approach is mixed with back-stepping control algorithm to compensate in the control design for model parameter uncertainty [5, 4]. Visual servo controllers were developed in [30, 31] to relieve the requirement of velocity measurement in application that only has VTOL position measurement via vision-based sensors.

2.1.2 Contribution

A simplified planar case MOVA system and the necessary steps for designing an autonomous controller to realize trajectory tracking control of the end-effector are presented in this chapter. Discussion of the modeling and control of MOVA system

is based on a simple and general MOVA design in 2D space. The VTOL aircraft is abstracted by a rigid body with a body-fixed vector thrust input and a torque input. The onboard manipulator has one revolute joint to provide the ability of placing the end-effector in arbitrary pose while the under-actuated VTOL aircraft is flying. Dynamics and control of a planar MOVA system is studied as a precursor of the investigation of the general three dimensional system. Dynamics model of the planar MOVA is derived and then transformed into a form suitable for controller design. Resulting equation of motion is validated against commercial numerical rigid dynamics simulation package. Controller of the planar system is designed based on the dynamics using back-stepping method. Stability of the proposed controller is studied using Lyapunovs stability analysis. Numerical simulation and a physical test-bed are constructed in order to evaluate the performance of the controller and both yield satisfying results. Although the outcome of the planar case study do not offer much value in implementation of a real MOVA system capable of navigation in the three dimensional physical world, it serves as a guidance for the later research of the 3D system. Moreover, the study of the planar case is considerably more concise than the full three-dimensional version and is thus recommended for reading as an overview of the entire dissertation.

2.1.3 Organization

In Sec. 2.2, a layout of the planar system is presented and kinematics, as well as the steps for constructing the alternative virtual manipulator kinematics, are presented. In Sec. 2.3, the dynamics model of planar MOVA is derived using the Lagrangian approach and then rewritten into a form suitable for an integrated controller design using the kinematics equations from virtual manipulator method. Simulation

of the derived dynamics equation is compared with commercial numerical dynamics simulation software package developed by third-party with same input for validation purpose. Back-stepping end-effector trajectory tracking controller is developed with inspiration from VTOL aircraft control in Sec. 2.3. The proposed controller is then proved to be GUUB using Lyapunov’s direct method. Simulation of the back-stepping trajectory tracking controller is displayed with satisfying result to show the effectiveness of the proposed controller and to promote future work on the three-dimension work and hardware-in-the-loop test-bed construction.

2.2 Kinematics of the Planar MOVA

This section describes a planar MOVA system structure and a derivation of the kinematic model using a virtual manipulator formation. The kinematic equations then form the basis for the equations of motion derived in Sec. 2.3.

2.2.1 System Structure

Design of a planar MOVA system begins with a two-dimensional (2D) VTOL aircraft. The VTOL aircraft is abstracted as a rigid body with one body-fixed force actuator and one torque actuator (see Fig. 2.1). The force actuator, shown as F in Fig. 2.1, is placed parallel to the “up” direction of the aircraft to simulate the vertical thrust generated by the VTOL aircraft. The torque, τ_0 , represents the attitude maneuvering capability of the VTOL and acts in the counter-clockwise in direction relative to the perpendicular to the x-z plane at the center of mass. Note that although the aircraft body has three degrees-of-freedom alone, the rotation is used for steering of the body-fixed thrust vector in order to achieve translation movement of the VTOL aircraft in the plane. Thus, only two independent degrees-of-freedom

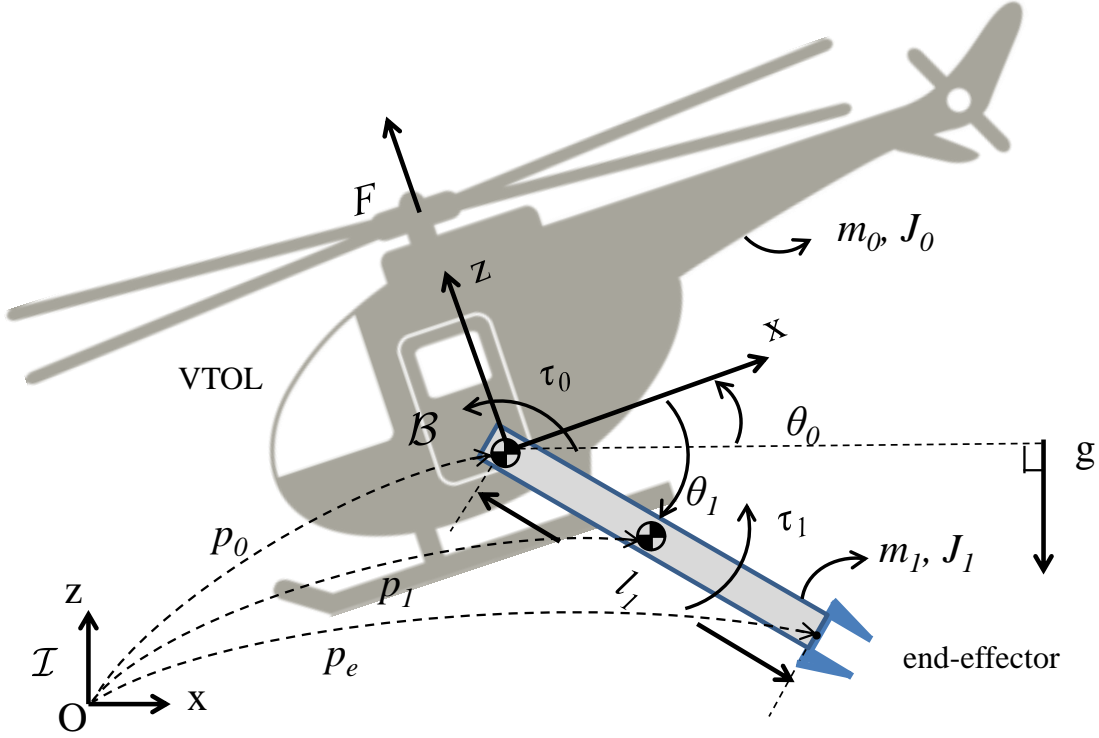


Figure 2.1: Coordinate system definition for the planar MOVA system. \mathcal{I} indicates the earth fixed coordinate frame and \mathcal{B} represents body-fixed coordinate frame.

are considered as controllable on the aircraft, which corresponds to the number of actuators on the aircraft body.

The design of the planar MOVA system proceeds by identifying the minimum number of degrees of freedom needed in the manipulator. The combined VTOL aircraft and the manipulator arm will move in the same x-z plane as the task object and hence requires that the 2DOF VTOL base be augmented by a manipulator with one degree-of-freedom thus producing three degrees-of-freedom of manipulation in task space, two from translation and one from rotation. That is, to achieve trajectory control of the planar MOVA end-effector, a three degree-of-freedom objective, one independent actuator must be added through the manipulator sub-system. The onboard manipulator is designed in the most weight frugal fashion without compro-

Table 2.1: Notation description

Symbol	Description
m_0	mass of the VTOL aircraft
J_0	moment of inertia of the VTOL aircraft
m_1	mass of the single manipulator link
J_1	moment of inertia of the manipulator link about its center of mass
l_1	length of the single manipulator link
x_0, z_0	coordinates of the center of mass of the VTOL aircraft
θ_0	attitude angle of the VTOL aircraft, with $\theta_0 = 0$ representing horizontal position
θ_1	angle formed between the VTOL aircraft and the manipulator link, with $\theta_1 = -\theta_0$ defined when the manipulator is pointing in the $+x$ direction
F	body-fixed thrust force generated by the VTOL aircraft
τ_0	torque on the body of the VTOL aircraft
τ_1	torque driving the single manipulator link

mising the control objective; that is, one single rotational joint is needed to augment the 2DOF body motion to create 3DOF positioning of the end-effector. The manipulator is attached to the center of mass of the aircraft body in order to minimize reaction torques. Additional joints may be added for a redundant system design; however, redundancy in kinematics will introduce issues as well as benefits and such a system is not considered.

Leveraging the simplicity of the planar model, variables describing the states of the MOVA system are represented in the inertial frame \mathcal{I} instead of the aircraft body frame \mathcal{B} as is typically done with regular (in three-dimensional space) aircraft models. This modeling decision will simplify the model development for the planar case, the full dimensional system will require body-fixed coordinates. Notation defining the constants and variables of the MOVA system are listed in Table 2.1.

2.2.2 Kinematics Derivation

As a starting point for the derivation of the MOVA system kinematics, a purely geometric analysis is made based on Fig. 2.1 to write the coordinates of several important points, which include center of mass of the VTOL, center of mass of the manipulator arm, and the point that represents the end-effector, in the inertial frame \mathcal{I} . A set of intuitive configuration variables, x_0 , z_0 , θ_0 and θ_1 (defined in Table 2.1) are used for representation of position and orientation.

The center of mass of the VTOL aircraft is denoted by the vector $p_0 = [x_0, z_0]^\top$.

The vector

$$p_1 = \begin{bmatrix} x_0 + \frac{1}{2}l_1 \cos(\theta_{01}) \\ z_0 + \frac{1}{2}l_1 \sin(\theta_{01}) \end{bmatrix} \quad (2.1)$$

locates the center of mass of the manipulator link and

$$p_e = \begin{bmatrix} x_0 + l_1 \cos(\theta_{01}) \\ z_0 + l_1 \sin(\theta_{01}) \end{bmatrix} \quad (2.2)$$

represents the geometric termination of the end-effector. The short-hand notation $\theta_{01} = \theta_0 + \theta_1$ is introduced to represent end-effector orientation as a sum of the tilt angle of the VTOL and manipulator joint angle.

The variable $p_e(p_0, \theta_0, \theta_1)$ describes the kinematic relationship between the VTOL position and orientation and the manipulator joint angle to the end-effector position.

This kinematics derivation is an immediate result of the geometric description of the VTOL aircraft and the attached manipulator. The kinematics chain starts from origin of the inertial frame \mathcal{I} and goes to the end-effector through the VTOL center of mass, p_0 , and each manipulator link (in the planar case only one link is

used). However, vector p_e is dynamically affected by both external forces and torques (the thrust force and torque on the VTOL aircraft) and internal forces and torques (the force and torque between each link and the VTOL aircraft body). This will be more obvious in Sec. 2.3 by inspecting terms in the derived dynamics equation (2.14). It is desired to represent the kinematics of the planar MOVA, p_e , in a decoupled form that is the sum of two parts: one contains only configuration variables related to the onboard manipulator and the other includes the rest. The decoupled form will yield more concise derivation of dynamics and offer conveniences in controller design for the MOVA system.

One general method for achieving this goal is to write the kinematics as a summation of a series of vectors that starts from the center of mass of the entire multi-body system called the virtual ground [16]. This approach is equivalent to finding an alternative kinematics chain that starts from origin of the inertial frame and ends at the end-effector, but goes through the center of mass of the entire MOVA system, i.e. the virtual ground. The series of vectors, which is together referred as the virtual manipulator, can be constructed using the virtual manipulator approach. The hypothetical virtual manipulator (see Fig. 2.2) has its first link located at the virtual ground. Derivation of the appropriate virtual manipulator guarantees that given the same joint angle configurations, the resulting virtual manipulator reaches the exact same end-effector position and orientation as the real manipulator.

For convenience, the links of both the original and virtual manipulator are one-base indexed; that is, for manipulator with N links, its links are numbered 1 to N , and the corresponding virtual manipulator links are numbered 1 to $N + 1$. Note that the corresponding virtual manipulator has one more link than the original manipulator. The added link, noted as Link 1, represents the attitude of the base of the original manipulator, or the VTOL aircraft in a MOVA system. In other words,

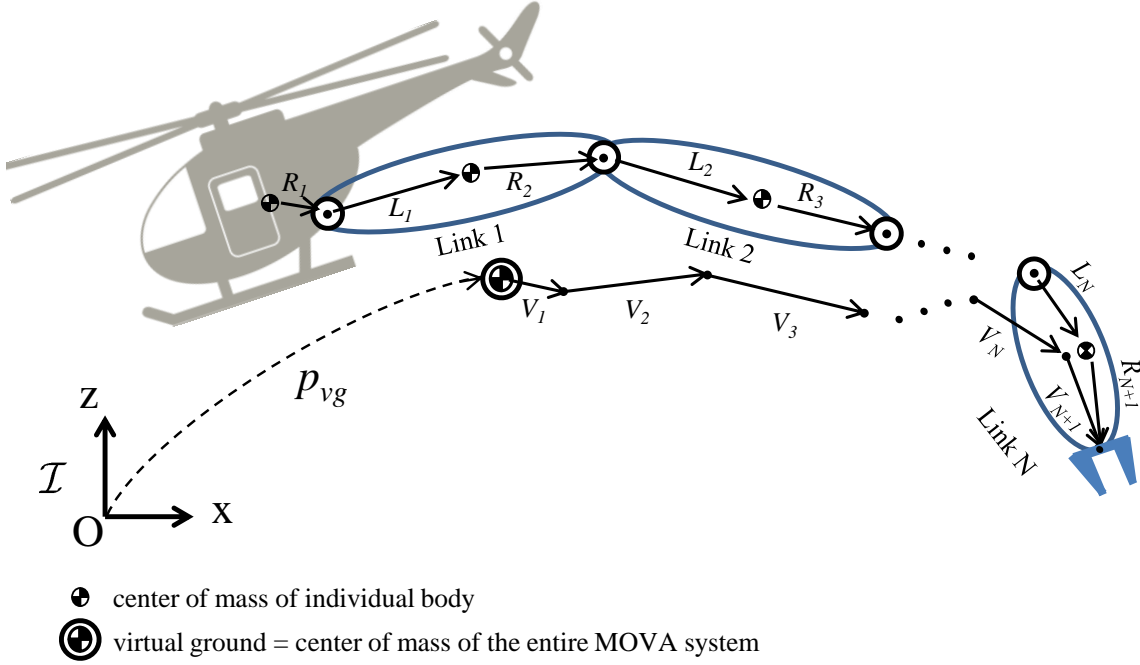


Figure 2.2: Real manipulator kinematics chain versus virtual manipulator kinematics chain for a general N-link MOVA system. Both chains reach the same endpoint with the same orientation.

Link $i + 1$ in virtual manipulator corresponds to Link i of the original manipulator.

The construction of a virtual manipulator is illustrated in two steps for general mobile manipulators. The i th virtual manipulator link in Fig. 2.2 is denoted by vector V_i , which is a sum of intermediate vectors D_i and H_i

$$\begin{aligned}
 V_1 &= D_1, \\
 V_i &= D_i + H_{i-1}, \quad (i = 2, 3 \dots N).
 \end{aligned}
 \tag{2.3}$$

The intermediate vectors D_i and H_i are defined by scaling the vectors R_i and L_i of

the physical manipulator, respectively, according to

$$\begin{aligned} D_i &= R_i \sum_{j=1}^i \frac{m_j}{m_T}, \quad (i = 1, 2 \dots N + 1) \\ H_i &= L_i \sum_{j=1}^i \frac{m_j}{m_T}, \quad (i = 1, 2 \dots N) \end{aligned} \tag{2.4}$$

in which m_j is the mass of j th link and $m_T = \sum_j m_j$ is the total mass of the system. Terms R_i and L_i are vectors defined in the original manipulator, representing displacements between Joint i and the center of mass of the two bodies it connects. Specifically R_i is the vector from center of mass of the previous body in the kinematic chain to Joint i and L_i is the vector from Joint i to the center of mass of the next body in the kinematic chain (see Fig. 2.2).

The starting point of the virtual manipulator, the virtual ground, is the center of mass of the entire multi-body system, which is a weighted average of vectors representing center of mass of each component $p_v g = \sum_i m_i p_i$. For the MOVA system, $p_v g = \sum_{i=0}^N m_i p_i$, where m_0 and p_0 are the mass and center of mass of the VTOL aircraft and m_i and p_i ($i = 1, 2 \dots N$) are mass and center of mass of the i th link of the onboard manipulator.

The relationship between rotation of virtual manipulator Link i and the corresponding original manipulator link, Link $i - 1$, is shown in Fig. 2.3. By convention, L_i and R_{i+1} are vectors in Link i and thus can be represented as constant body-fixed vectors on Link i of the original manipulator. From (2.4), D_i and H_i are R_i and L_i scaled by constants, respectively. Thus, H_i and D_{i+1} can be also written as constant body-fixed vectors on Link i of the original manipulator. By definition of V_i in (2.3), $V_{i+1} = D_{i+1} + H_i$ is a constant body-fixed vector on Link i of original manipulator as well. Thus, rotation of Link i in the original manipulator is equivalent to rotation

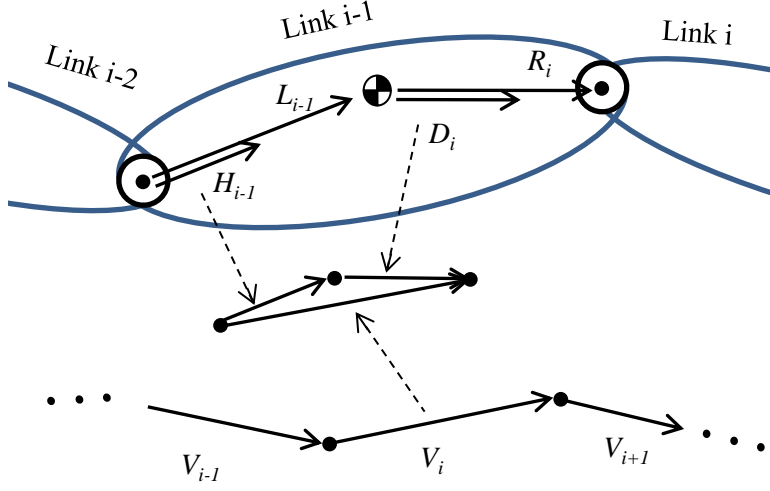


Figure 2.3: View of a single link of the virtual manipulator and original manipulator. The corresponding links never rotate relative to each other.

of Link $i + 1$ of the virtual manipulator and the rotation of the base of the original manipulator (the VTOL body for the MOVA) is equivalent to rotation of Joint 1 of virtual manipulator.

For the specific planar MOVA system with one rotational link discussed in this section, $N = 1$. Vectors L_i and R_i are calculated from the system geometry in Fig. 2.1

$$\begin{aligned}
 R_1 &= [0, 0]^T, \\
 L_1 &= p_1 - p_0 = \frac{l_1}{2} \begin{bmatrix} \cos \theta_{01} \\ \sin \theta_{01} \end{bmatrix}, \\
 R_2 &= p_e - p_1 = \frac{l_1}{2} \begin{bmatrix} \cos \theta_{01} \\ \sin \theta_{01} \end{bmatrix}.
 \end{aligned} \tag{2.5}$$

Note $R_1 = [0, 0]^T$ because the manipulator Joint 1 and the center of mass of the

VTOL aircraft are co-located. By (2.4), vectors D_1 , H_1 and D_2 are derived

$$\begin{aligned} D_1 &= [0, 0]^\top, \\ H_1 &= \frac{m_0}{m_T} L_1 = \frac{l_1 m_0}{2m_T} \begin{bmatrix} \cos \theta_{01} \\ \sin \theta_{01} \end{bmatrix} \\ D_2 &= R_2 = \frac{l_1}{2} \begin{bmatrix} \cos \theta_{01} \\ \sin \theta_{01} \end{bmatrix}, \end{aligned} \tag{2.6}$$

where $m_T = m_0 + m_1$.

The vectors representing the links of the virtual manipulator, calculated following (2.3), are

$$\begin{aligned} V_1 &= D_1 = [0, 0]^\top, \\ V_2 &= D_2 + H_1 = \frac{l_1 (m_T + m_0)}{2m_T} \begin{bmatrix} \cos \theta_{01} \\ \sin \theta_{01} \end{bmatrix}. \end{aligned} \tag{2.7}$$

The coordinates of the virtual ground in frame \mathcal{I} is calculated by averaging p_0 and p_1 weighted with masses of VTOL aircraft and Link 1 of the manipulator respectively

$$\begin{aligned} p_{vg} &= \sum_{i=0}^N m_i p_i \\ &= \frac{1}{m_T} (m_0 p_0 + m_1 p_1) \\ &= p_0 + \frac{l_1 m_1}{2m_T} \begin{bmatrix} \cos \theta_{01} \\ \sin \theta_{01} \end{bmatrix}. \end{aligned} \tag{2.8}$$

The corresponding velocity vector

$$v_{vg} = v_0 + \frac{l_1 m_1}{2m_T} \dot{\theta}_{01} \begin{bmatrix} -\sin \theta_{01} \\ \cos \theta_{01} \end{bmatrix}, \quad (2.9)$$

where $v_0 = \dot{p}_0 = [\dot{x}_0, \dot{z}_0]^\top$.

Using the virtual manipulator representation, the kinematics of the planar MOVA system is re-derived. The position of the end-effector can be written in the form

$$\begin{aligned} p_e &= p_{vg} + V_1 + V_2 \\ &= p_{vg} + \frac{l_1 (m_T + m_0)}{2m_T} \begin{bmatrix} \cos \theta_{01} \\ \sin \theta_{01} \end{bmatrix}, \end{aligned} \quad (2.10)$$

and the orientation of the end-effector has its original representation θ_{01} . Further substituting (2.8) into (2.10) will yield a vector identical to p_e in its original form given in (2.2), meaning the virtual manipulator approach results in an alternative kinematics for the end-effector of the planar MOVA system. Vector p_{vg} is only influenced by external forces and torques and is thus dynamically decoupled from the onboard manipulator.

In summary, the kinematics of the planar MOVA system is derived from geometry of system. Then, the kinematic chain of the end-effector is transformed into a representation with virtual manipulator approach. The new representation involves a virtual manipulator extended from virtual ground (center of mass of the entire MOVA system) instead of a point fixed on the VTOL aircraft. The virtual ground point has the property of not being affected by internal forces and torques in the multi-body system, which will help to yield a decoupled dynamics and facilitate control design.

2.3 Dynamics of the Planar MOVA

In this section, the dynamics of the planar MOVA system will be derived and verified through simulation. The MOVA dynamics are first modeled with a Lagrangian approach using the general coordinate vector

$$q = \begin{bmatrix} x_0 \\ z_0 \\ \theta_0 \\ \theta_1 \end{bmatrix} \quad (2.11)$$

and then rewritten with respect to the virtual manipulator representation. Later, the dynamics equation is validated by comparing simulation results with the results produced from the SimMechanics[®] numerical multi-body dynamics package.

2.3.1 Dynamics by Euler-Lagrangian Approach

The kinetic energy T and potential energy V of the MOVA system are

$$\begin{aligned} T &= \frac{1}{2}m_0\dot{p}_0^\top\dot{p}_0 + \frac{1}{2}m_1\dot{p}_1^\top\dot{p}_1 + \frac{1}{2}J_0\dot{\theta}_0^2 + \frac{1}{2}J_1\dot{\theta}_{01}^2, \\ V &= [0, m_Tg] \cdot p_{vg} = g \left(m_Tz_0 + \frac{m_1}{2}l_1 \sin \theta_{01} \right), \end{aligned} \quad (2.12)$$

and the Lagrangian is $L = T - V$. Following the standard Euler-Lagrangian approach, the equation of motion of the MOVA system under the general coordinate q is found

by evaluating

$$\begin{aligned}
\frac{d}{dt} \frac{\partial L}{\partial \dot{x}_0} - \frac{\partial L}{\partial x_0} &= F_x, \\
\frac{d}{dt} \frac{\partial L}{\partial \dot{z}_0} - \frac{\partial L}{\partial z_0} &= F_z, \\
\frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_0} - \frac{\partial L}{\partial \theta_0} &= \tau_0, \\
\frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_1} - \frac{\partial L}{\partial \theta_1} &= \tau_1,
\end{aligned} \tag{2.13}$$

where $F_x = -F \sin \theta_0$ and $F_z = F \cos \theta_0$ are projections of the thrust force F attached to the VTOL aircraft onto the x- and z-axis in \mathcal{I} . The resulting dynamics equation can be put into the form

$$M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + G(q) = \tau \tag{2.14}$$

where $M \in \mathbf{R}^{4 \times 4}$ is the inertia matrix, $C \in \mathbf{R}^{4 \times 4}$ is a matrix containing nonlinear centripetal and Coriolis terms, $G \in \mathbf{R}^{4 \times 1}$ represents gravity effects, and $\tau \in \mathbf{R}^{4 \times 1}$ is the general force vector. The complete matrices and vectors $M(q)$, $C(q, \dot{q})$, $G(q)$, τ are given by

$$M = \begin{bmatrix} m_T & 0 & -\frac{1}{2}l_1m_1 \sin \theta_{01} & -\frac{1}{2}l_1m_1 \sin \theta_{01} \\ 0 & m_T & \frac{1}{2}l_1m_1 \cos \theta_{01} & \frac{1}{2}l_1m_1 \cos \theta_{01} \\ -\frac{1}{2}l_1m_1 \sin \theta_{01} & \frac{1}{2}l_1m_1 \cos \theta_{01} & J_0 + J_1 + \frac{l_1^2m_1}{4} & J_1 + \frac{l_1^2m_1}{4} \\ -\frac{1}{2}l_1m_1 \sin \theta_{01} & \frac{1}{2}l_1m_1 \cos \theta_{01} & J_1 + \frac{l_1^2m_1}{4} & J_1 + \frac{l_1^2m_1}{4} \end{bmatrix}, \tag{2.15}$$

$$C = \begin{bmatrix} 0 & 0 & -\frac{1}{2}l_1m_1 \cos \theta_{01} \dot{\theta}_{01} & -\frac{1}{2}l_1m_1 \cos \theta_{01} \dot{\theta}_{01} \\ 0 & 0 & -\frac{1}{2}l_1m_1 \sin \theta_{01} \dot{\theta}_{01} & -\frac{1}{2}l_1m_1 \sin \theta_{01} \dot{\theta}_{01} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad (2.16)$$

$$G = \begin{bmatrix} 0 & m_T g & \frac{1}{2}l_1m_1 \cos \theta_{01} g & \frac{1}{2}l_1m_1 \cos \theta_{01} g \end{bmatrix}^T, \quad (2.17)$$

$$\tau = \begin{bmatrix} -F \sin \theta_0 & F \cos \theta_0 & \tau_0 & \tau_1 \end{bmatrix}^T. \quad (2.18)$$

Notice that VTOL aircraft translational (first two rows), VTOL aircraft rotational (third row), and manipulator link (last row) dynamics are coupled via off-diagonal entries in $M(q)$ and $C(q, \dot{q})$ in (2.15) and (2.16). If these off-diagonal terms were always zero, the system could be split into several subsystems and controlled individually without causing control performance issues. Applying a separate control strategy when this condition is not satisfied is equivalent to designing controllers for a system with off-diagonal terms removed. The resulting closed-loop system will sustain disturbances caused by the uncompensated dynamics, which may lead to deteriorated performance.

The potential effect of the coupling terms for the planar MOVA system can be estimated by considering the first row of $M(q)\ddot{q}$, which equals $m_T\ddot{x}_0 - \frac{1}{2}l_1m_1 \sin \theta_{01}\ddot{\theta}_{01}$, where $\ddot{\theta}_{01} = \ddot{\theta}_0 + \ddot{\theta}_1$. For given manipulator motions, the size of l_1m_1 relative to m_T will dictate the impact of the disturbance. For a short and light manipulator on a large VTOL aircraft ($l_1m_1 \ll m_T$), it is likely that the disturbances will be small. In the case where $l_1m_1 = O(m_T)$, which is the scenario proposed in this work, the disturbance cannot be neglected without consequence and an integrated controller is

required. The back-stepping controller design approach will be used to obtain such an integrated controller but the approach requires the system dynamics in a strict-feedback form. The off-diagonal terms in $M(q)$ indicate the system is not currently in that form and will be an obstacle to control design. Thus, a transform into a decoupled dynamics representation is needed.

2.3.2 Decoupling of Translational Dynamics

From the derivation in Sec. 2.2.2, the kinematics of the planar MOVA can be rewritten following the virtual manipulator representation into a kinematics chain passing through the virtual ground, which is also the center of mass of the entire MOVA system. Position of the virtual ground is only affected by external force inputs, simply thrust force in the planar MOVA system. Rewriting the translational dynamics of the planar MOVA system in (2.14) in terms of p_{vg} will decouple it from rotation of the VTOL and the manipulator.

The first two rows of the dynamics equation (2.14) are extracted as

$$\begin{bmatrix} m_T & 0 \\ 0 & m_T \end{bmatrix} \ddot{p}_0 + \begin{bmatrix} -\frac{1}{2}l_1m_1 \sin \theta_{01} \ddot{\theta}_{01} - \frac{1}{2}l_1m_1 \cos \theta_{01} \dot{\theta}_{01}^2 \\ \frac{1}{2}l_1m_1 \cos \theta_{01} \ddot{\theta}_{01} - \frac{1}{2}l_1m_1 \sin \theta_{01} \dot{\theta}_{01}^2 + m_T g \end{bmatrix} = \begin{bmatrix} -F \sin \theta_0 \\ F \cos \theta_0 \end{bmatrix}, \quad (2.19)$$

where $\ddot{\theta}_{01} = \ddot{\theta}_0 + \ddot{\theta}_1$ and $\dot{\theta}_{01} = \dot{\theta}_0 + \dot{\theta}_1$. Taking the time derivative of (2.9) yields,

$$\ddot{p}_{vg} = \ddot{p}_0 + \frac{l_1m_1}{2m_T} \left(\begin{bmatrix} -\sin \theta_{01} \\ \cos \theta_{01} \end{bmatrix} \ddot{\theta}_{01} - \begin{bmatrix} \cos \theta_{01} \\ \sin \theta_{01} \end{bmatrix} \dot{\theta}_{01}^2 \right). \quad (2.20)$$

The decoupled translational dynamics of p_{vg} are obtained by multiplying $m_T I_2$ (I_2 is 2×2 identity matrix) into both sides of equation (2.20) then substituting result into

(2.19) to yield

$$m_T \ddot{p}_{vg} + \begin{bmatrix} 0 \\ m_T g \end{bmatrix} = \begin{bmatrix} -F \sin \theta_0 \\ F \cos \theta_0 \end{bmatrix}. \quad (2.21)$$

This result is equivalent to Newton's law on the center of mass of the entire system. Notice that the dynamics of p_{vg} , the center of mass of the planar system, appears in a form that is similar to a VTOL aircraft in 2D space. We can exploit this fact and reference VTOL control research when designing a controller for p_{vg} .

Concise representation of the VTOL aircraft rotation and end-effector orientation dynamics can be obtained by manipulating selected rows in (2.14). A simpler form of θ_0 dynamics is found by subtracting row four from row three of (2.14), which yields

$$J_0 \ddot{\theta}_0 = \tau_0 - \tau_1. \quad (2.22)$$

Orientation of the end-effector is $\theta_{01} = \theta_0 + \theta_1$, for which dynamics can be found by rewriting row four of (2.14) to show

$$\frac{1}{2} l_1 m_1 [-\sin \theta_{01}, \cos \theta_{01}] \ddot{p}_0 + \left(J_1 + \frac{l_1^2 m_1}{4} \right) \ddot{\theta}_{01} + \frac{1}{2} l_1 m_1 \cos \theta_{01} g = \tau_1. \quad (2.23)$$

Substituting \ddot{p}_0 of (2.19) into (2.23), θ_{01} dynamics is obtained after reducing terms by trigonometric identities and reorganizing,

$$\left(J_1 + \frac{l_1^2 m_0 m_1}{4 m_T} \right) \ddot{\theta}_{01} + \frac{1}{2} \frac{l_1 m_1}{m_T} \cos \theta_1 F = \tau_1, \quad (2.24)$$

which can be written in a form

$$\bar{J}_1 \ddot{\theta}_{01} = \tau_1 - \xi_1 F, \quad (2.25)$$

which incorporates the short-hand notation $\bar{J}_1 = J_1 + \frac{l_1^2 m_0 m_1}{4m_T}$ and $\xi_1 = \frac{l_1 m_1}{2m_T} \cos \theta_1$.

At this point, the dynamics of the system described by (2.14) is completely rewritten into the form in (2.21), (2.22), and (2.25). These dynamics equations will be used as the model for the planar MOVA in design of a controller.

2.3.3 Validation of dynamics equation

The dynamics equations of the planar MOVA system are the basis for later model-based controller development. Although above they are decoupled into a more friendly form, the transformation is built upon the original complicated form in (2.14). Minor errors are possible during the Lagrangian partial derivative calculation and propagate into the final decoupled form. Note that these possible errors are likely not to have any effect on the controller design and simulation of the closed-loop system will not show any sign of abnormality as the closed-loop dynamics may not even contain the incorrectly evaluated term (due to cancellation by the controller). However, these errors will affect the performance on a real physical system as they are essentially uncompensated by controller. Thus, it is necessary to validate the derived dynamics equation before proceeding. A truly “independent” computer simulation is proposed to validate the dynamics equations.

Numerical multi-body dynamics simulation software is used to provide a ground truth. These software packages take geometric constraints and the body physical properties of the system as input and generate a numerical representation of the dynamics system behavior under a certain initial condition and input. Although these simulations does not replace analytical dynamics derivation as it does not yield a closed-form representation of dynamics, it can be very helpful in validation of the derived dynamics.

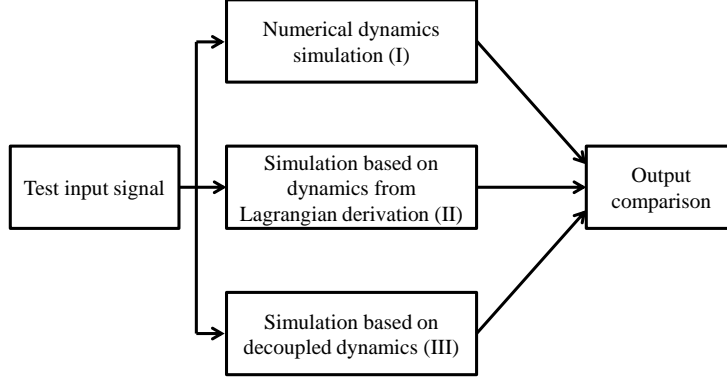


Figure 2.4: Block diagram of dynamics validation test-bed.

The elements of the validation test-bed are illustrated in Fig. 2.4. Test input signal is fed into three different simulations, first of which is based on a numerical dynamics simulation (I); the second is based on the dynamics derived via the Euler-Lagrangian approach (2.14) (II); the last is based on the decoupled dynamics equations (2.19),(2.22) and (2.25) (III). By comparing the output of the three simulations, fidelity of the derived model can be evaluated.

SimMechanics[®] toolbox of Simulink[®] was used as the numerical dynamics simulator [21]. Validation result of one of the instance is shown in Fig. 2.5, in which difference between end-effector positions from three simulations are used for comparison. The SimMechanics[®] simulation (I) output is chosen as the ground truth, since it has been developed by a third-party and has been used for years in practice by many users. End-effector position from the Lagrangian dynamics equation simulation (II) and decoupled dynamics equation simulation (III), which are both carried out using Simulink[®] software, are subtracted from the ground truth to form two errors, $e_{II}(t)$ and $e_{III}(t)$. The norm of two errors are summed up to create the cumulative yield

Table 2.2: Simulation Plant Physical Parameters

Parameter	Value
m_0	1 kg
J_0	0.125 kg·m
m_1	0.5 kg
J_1	0.031 kg·m
l_1	0.5 m

simulation error,

$$\text{simulation error}(t) = e_{II}(t) + e_{III}(t) \quad (2.26)$$

a scalar value that indicate the closeness of three simulation outputs. Moreover, the relative error is quantified by the ratio between the simulation error and the norm of ground truth end-effector position vector. Physical parameters of the planar MOVA system used in the simulation are listed in Table 2.2, these parameters approximate a small quadrotor helicopter. Note that the mass and moment of inertia of the VTOL aircraft body and the one-link manipulator are comparable as previously assumed. For all tested below, the initial condition is set as $p_0(0) = [0, 0]^T$, $\theta_0(0) = \theta_1(0) = 0$.

In the first test case, a random noise signal was used as the test signal. All three input signals, $F(t)$, $\tau_0(t)$, and $\tau_1(t)$, are piece-wise constant with values determined by the corresponding discrete time signals, $F[n]$, $\tau_0[n]$, and $\tau_1[n]$, with sample time $T_s = 0.1$ sec,

$$\begin{bmatrix} F(t) \\ \tau_0(t) \\ \tau_1(t) \end{bmatrix} = \begin{bmatrix} F[n] \\ \tau_0[n] \\ \tau_1[n] \end{bmatrix} \quad (nT_s \leq t < (n+1)T_s). \quad (2.27)$$

Discrete time signals $F[n]$, $\tau_0[n]$, and $\tau_1[n]$ follow uniform distributions with zero means, $F[n] \sim U(-10, 10)$, $\tau_0[n], \tau_1[n] \sim U(-1, 1)$. The simulation was run for 10 sec

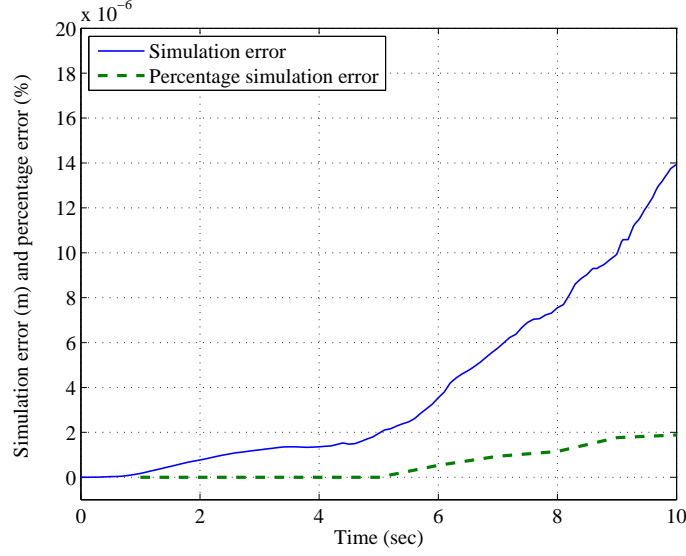


Figure 2.5: Plot of dynamics validation result with random input signals.

and the simulation error shown in Fig. 2.5 is on the order of 1×10^{-5} m and the relative error of $1 \times 10^{-6}\%$, which is sufficiently accurate for controller design.

In the second case, sine wave signals are fed into the simulations to show the modeled planar MOVA system acts as expected. The input signals, $F(t)$, $\tau_0(t)$, and $\tau_1(t)$, are generated by

$$\begin{bmatrix} F(t) \\ \tau_0(t) \\ \tau_1(t) \end{bmatrix} = \begin{bmatrix} m_T g + 3 \sin 10t \\ \sin 5t \\ \sin 5t \end{bmatrix}. \quad (2.28)$$

Note that $\tau_0(t) = \tau_1(t)$, which should result in a constant VTOL aircraft attitude by dynamic equation (2.22). The simulation was run for 10 sec and the simulation error shown in 2.6. The relative error is on the order of $1 \times 10^{-4}\%$, larger than that in the previous case, because the norm of p_e remains relatively small during the simulation (see plot of p_e in Fig. 2.7).

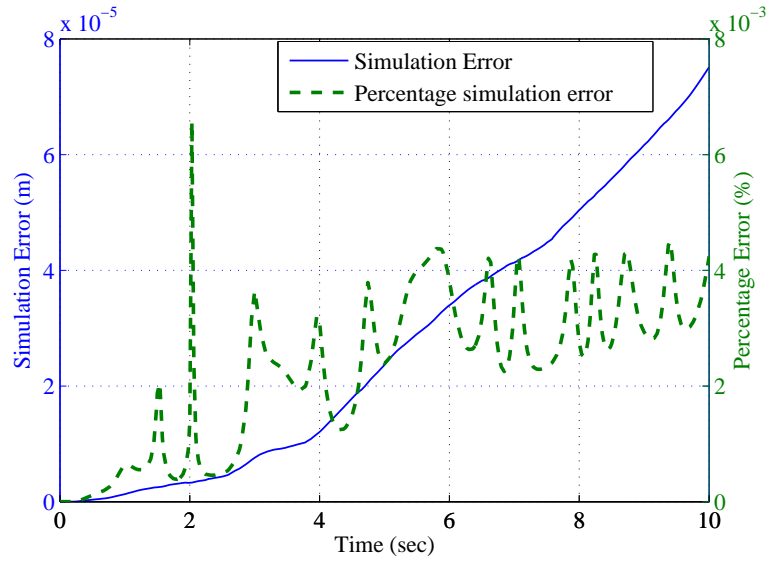


Figure 2.6: Plot of dynamics validation result with the sine wave test input signal.

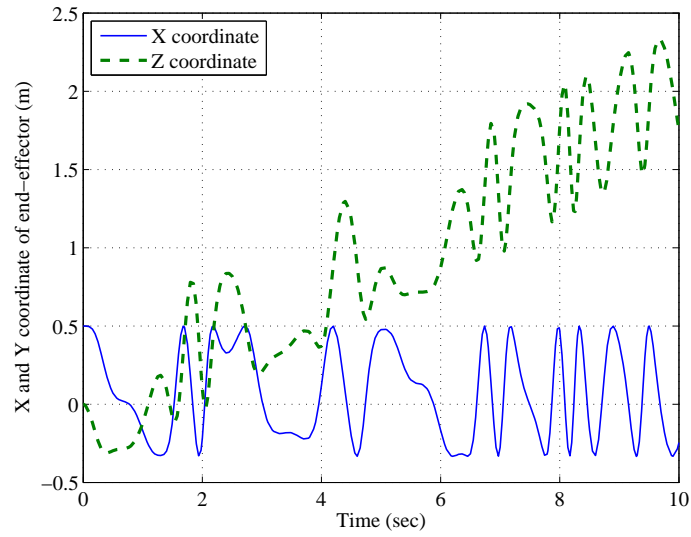


Figure 2.7: Plot of p_e with the sine wave test input signal.

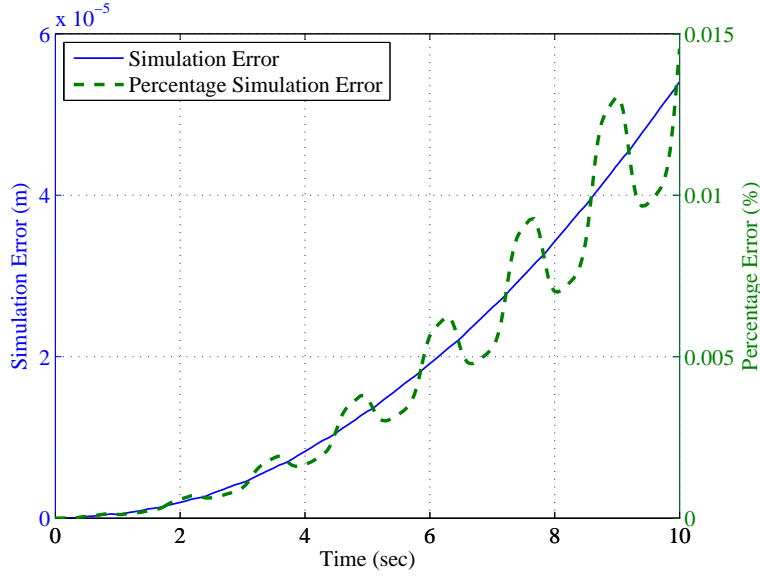


Figure 2.8: Plot of dynamics validation result with the constant test input signal.

In the third case, the input signals are kept constant. Input signals, $\tau_0(t)$, and $\tau_1(t)$ are set to zero. The only non-zero input $F(t) = m_T g$ perfectly balances the gravity force. Thus, the center of mass of the entire system is expected remain at the same location. The manipulator link oscillates with constant amplitude like a pendulum because the simulation starts with $\theta_{01} = 0$, a condition away from the stable equilibrium point $\theta_{01} = \pi/2 + 2n\pi (n \in \mathbf{Z})$, and the link natural damping is zero. The simulation was run for 10 sec and the simulation error shown in Fig. 2.8. The relative error is on the order of $1 \times 10^{-2}\%$, larger than in the random test input signal case for the same reason as the second case. The plot of p_e is shown in Fig. 2.9, in which constant amplitude oscillation in both x and y coordinates is observed. The coupled movement of the VTOL in the x-direction in response to arm motions highlights the need for control of the full-order, coupled MOVA system.

From all three validation cases, simulation errors and relative simulation errors remain small, which indicates that the derived dynamic models are close to the assumed ground truth and the two derived model are close to each other. Although

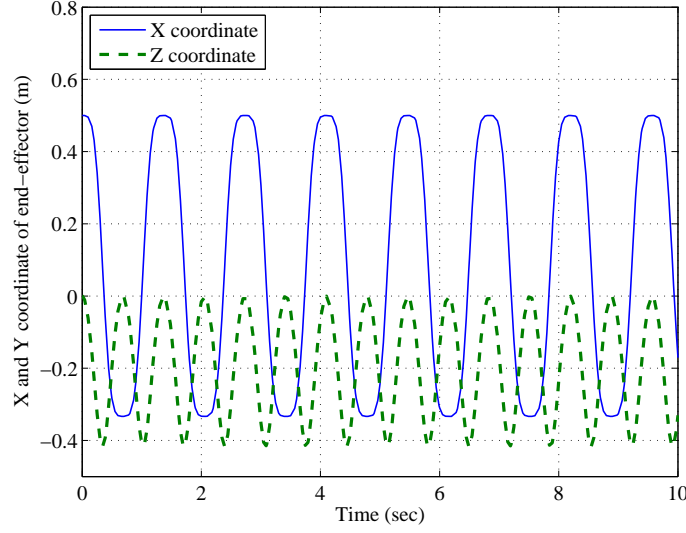


Figure 2.9: Plot of p_e with the constant test input signal.

non-zero, the error is small enough to conclude that the derived models capture the system behavior and are suitable for the following model-based controller design. The constant input case also shows the system heuristically behaves as expected. The source of error is not only related to the accuracy of model, but also in relation to the numerical dynamics simulator and overall simulation settings.

2.4 Controller Design for the Planar MOVA

In this section, a unified controller is designed to achieve trajectory tracking control of the end-effector of the planar MOVA system. The dynamics model used for controller formulation is detailed in (2.21), (2.22), and (2.25). The trajectory of virtual ground p_{vg} is derived from the desired position and orientation trajectory of the end-effector by applying the kinematics found in Sec. 2.2. Inspired by VTOL control research in [4] and [36], the back-stepping control design technique is adopted to control the underactuated p_{vg} dynamics. The end-effector orientation dynamics,

represented in (2.25), is driven by a control law derived using feedback linearization. Stability analysis is performed on the proposed control design and a GUUB tracking result is achieved. To demonstrate performance of the proposed controller, results from numerical simulation are presented with satisfactory end-effector position and orientation tracking error, which indicates a successful controller design.

The control objective is to force the end-effector position p_e and orientation θ_{01} , in the inertial frame \mathcal{I} , to converge to reference trajectories $p_{er}(t)$ and $\theta_{01r}(t)$, respectively, with additional subscript “r” denotes the reference version of the corresponding variable. The reference trajectories $p_{er}(t)$ and $\theta_{01r}(t)$ are assumed to be sufficiently smooth and up to the third time derivatives are bounded, i.e., $p_{er}(t)$, $\dot{p}_{er}(t)$, $\ddot{p}_{er}(t)$, and $\dddot{p}_{er}(t) \in \mathcal{L}_\infty$, $\theta_{01r}(t)$, $\dot{\theta}_{01r}(t)$, $\ddot{\theta}_{01r}(t)$, and $\dddot{\theta}_{01r}(t) \in \mathcal{L}_\infty$. Using the kinematics stated in (2.2) and (2.8), the virtual ground p_{vg} can be written in the form

$$p_{vg} = p_e - \frac{(m_T + m_0)l_1}{2m_T} \begin{bmatrix} \cos \theta_{01} \\ \sin \theta_{01} \end{bmatrix}. \quad (2.29)$$

Thus, the reference trajectory of the virtual ground, p_{vgr} , can be found by substituting $p_{er}(t)$ and $\theta_{01r}(t)$ to yield

$$p_{vgr} = p_{er} - \frac{(m_T + m_0)l_1}{2m_T} \begin{bmatrix} \cos \theta_{01r} \\ \sin \theta_{01r} \end{bmatrix}, \quad (2.30)$$

and \dot{p}_{vgr} , \ddot{p}_{vgr} and \dddot{p}_{vgr} can be found by taking time derivatives. The control design problem is then divided into two sub-problems: ensuring $p_{vg}(t)$ follows $p_{vgr}(t)$ and $\theta_{01}(t)$ follows $\theta_{01r}(t)$ by specifying appropriate input signals $F(t)$, $\tau_0(t)$ and $\tau_1(t)$, which represents VTOL thrust force, VTOL torque, and torque applied to the onboard manipulator, respectively.

2.4.1 Back-stepping Control of the Virtual Ground

The control input to the p_{vg} dynamics equation (2.21) represents a body-fixed thrust force vector, thus a tracking error system for control of p_{vg} is constructed in body frame \mathcal{B} for convenience. The position error, $e_p(t)$, in the body-fixed frame \mathcal{B} is defined as

$$e_p = R^\top (p_{vg} - p_{vgr}), \quad (2.31)$$

and the velocity error, $e_v(t)$ is defined in the same fashion as

$$e_v = R^\top (\dot{p}_{vg} - \dot{p}_{vgr}), \quad (2.32)$$

where $R \in SO(2)$ is the rotation matrix that transforms a vector in the inertial frame \mathcal{I} into the VTOL body frame \mathcal{B}

$$R = \begin{bmatrix} \cos \theta_0 & -\sin \theta_0 \\ \sin \theta_0 & \cos \theta_0 \end{bmatrix}. \quad (2.33)$$

Time derivatives of the position error e_p and the velocity error e_v are

$$\begin{aligned} \dot{e}_p &= S(\omega)e_p + R^\top (\dot{p}_{vg} - \dot{p}_{vgr}) \\ &= S(\omega)e_p + e_v, \\ \dot{e}_v &= S(\omega)e_v + R^\top (\ddot{p}_{vg} - \ddot{p}_{vgr}), \end{aligned} \quad (2.34)$$

where $\omega = \dot{\theta}_0$ and $S(\omega) \in so(2)$ is a skew-symmetric matrix

$$S(\omega) = \begin{bmatrix} 0 & \omega \\ -\omega & 0 \end{bmatrix}. \quad (2.35)$$

$S(\omega)$ is associated with R by $\frac{d}{dt}R = S(\omega)R$. Substituting the p_{vg} dynamics from (2.21) into \dot{e}_v in (2.34) yields

$$\begin{aligned}\dot{e}_v &= S(\omega)e_v - R^\top \ddot{p}_{vgr} - R^\top \left(\frac{F}{m_T} \begin{bmatrix} \sin \theta_0 \\ \cos \theta_0 \end{bmatrix} + \begin{bmatrix} 0 \\ g \end{bmatrix} \right) \\ &= S(\omega)e_v - R^\top (\ddot{p}_{vgr} + g_v) + B_F F,\end{aligned}\tag{2.36}$$

where the gravity force vector $g_v = [0, g]^\top$ and $B_F = [0, m_T^{-1}]^\top$ is the input matrix associated with input F .

A filtered tracking error $r(t)$ is defined as

$$r = e_v + \alpha e_p + \delta,\tag{2.37}$$

where $\alpha \in \mathbf{R}^+$ is a control gain and $\delta = [0, \delta_2]^\top \in \mathbf{R}^{2 \times 1}$, $\delta_2 > 0$, is an auxiliary vector formulated to introduce the opportunity to use back-stepping in the error dynamics. Dynamics of r can be found by taking the derivative of (2.37) and substituting in \dot{e}_p from (2.34) and \dot{e}_v from (2.36)

$$\dot{r} = S(\omega)e_v - R^\top (\ddot{p}_{vgr} + g_v) + B_F F + \alpha S(\omega)e_p + \alpha e_v.\tag{2.38}$$

The dynamics of r is underactuated, because equation (2.38) has two degrees-of-freedom while the control input F is a single thrust force actuator. An additional control input is desired to regulate the filtered tracking error r . Such an input can be indirectly created through ω , which coincides with the fact that VTOL aircraft steers the body-fixed thrust vector to achieve acceleration in different directions.

The effect of ω on the filtered tracking error is embedded in the $S(\omega)$ terms

in (2.38), which can be revealed by grouping terms related to r to yield

$$\begin{aligned}\dot{r} &= S(\omega)(e_v + \alpha e_p + \delta) + \alpha e_v - R^\top (\ddot{p}_{vgr} + g_v) + B_F F - S(\omega)\delta \\ &= S(\omega)r + \alpha e_v - R^\top (\ddot{p}_{vgr} + g_v) + B_F F - S(\omega)\delta.\end{aligned}\tag{2.39}$$

In the process, a $S(\omega)\delta$ is created to complete $S(\omega)r$, and the same term is subtracted at the end to maintain equality. Invoking the identity $S(\omega)\delta = [\delta_2, 0]^\top \omega$ and grouping both the input F and the ω into a vector form, gives

$$\begin{aligned}\dot{r} &= S(\omega)r + \alpha e_v - R^\top (\ddot{p}_{vgr} + g_v) + \begin{bmatrix} -\delta_2 & 0 \\ 0 & m_T^{-1} \end{bmatrix} \cdot \begin{bmatrix} \omega \\ F \end{bmatrix} \\ &= S(\omega)r + \xi_2 + B_\mu \mu,\end{aligned}\tag{2.40}$$

where $\xi_2 = \alpha e_v - R^\top (\ddot{p}_{vgr} + g_v)$,

$$B_\mu = \begin{bmatrix} -\delta_2 & 0 \\ 0 & m_T^{-1} \end{bmatrix}, \text{ and } \mu = \begin{bmatrix} \omega \\ F \end{bmatrix}.\tag{2.41}$$

It is clear that ω will affect the filtered tracking error r , however, $\omega = \dot{\theta}_0$ is not a direct control input, and can only be altered through θ_0 dynamics (2.22). Thus, μ cannot be directly specified to stabilize r as ω is one of its element.

The problems is abstracted as controlling the states of cascaded subsystems, where the desired control input is governed by its own dynamics. This is a typical situation for applying back-stepping technique. The process begins by first assuming that there is direct control over μ and continues to seek the desired value for μ , called

μ_d , to regulate r ; from (2.40), it is clear that

$$\mu_d = \begin{bmatrix} \omega_d \\ F_d \end{bmatrix} = B_\mu^{-1}(-k_r r - \xi_2 - e_p), \quad (2.42)$$

where ω_d and F_d are the two entries of μ_d would act to stabilize the r dynamics, the term $-k_r r$ is the regulating term, where $k_r = \text{diag}(k_{r1}, k_{r2}) \in \mathbf{R}^{2 \times 2}$ is a diagonal control gain matrix, $-\xi_2$ is for cancellation of unnecessary dynamics, and $-e_p$ is added to fulfill later stability analysis.

The mismatch, μ_e , between input μ and desired input μ_d is

$$\mu_e = \begin{bmatrix} \omega_e \\ F_e \end{bmatrix} = \mu - \mu_d = \begin{bmatrix} \omega - \omega_d \\ F - F_d \end{bmatrix} \quad (2.43)$$

In order to accomplish the goal of controlling r via μ , it is necessary that the control input μ approach μ_d as close as possible, which is the same as minimizing norm of μ_e by specifying the control input properly. For the directly controlled force input F , it is feasible to let

$$F = F_d = \begin{bmatrix} 0 & 1 \end{bmatrix} \mu_d, \quad (2.44)$$

which leads to $F_e = 0$. For the indirectly controlled input ω , the source of the control problem can be identified by looking at the dynamics of the mismatched error ω_e written as

$$\begin{aligned} \dot{\omega}_e &= \frac{d}{dt}(\omega - \omega_d) \\ &= \dot{\omega} - \begin{bmatrix} 1 & 0 \end{bmatrix} \dot{\mu}_d. \end{aligned} \quad (2.45)$$

Multiplying both side by J_0 and then substituting $\ddot{\theta}_0$ from (2.22) for $\dot{\omega}$, there is

$$\begin{aligned} J_0 \dot{\omega}_e &= J_0 \ddot{\theta}_0 - \begin{bmatrix} J_0 & 0 \end{bmatrix} \dot{\mu}_d \\ &= \tau_n - \begin{bmatrix} J_0 & 0 \end{bmatrix} \dot{\mu}_d, \end{aligned} \quad (2.46)$$

where $\tau_n = \tau_0 - \tau_1$ denotes the net torque acting on the VTOL aircraft. A control law for τ_n is designed by considering that ω_e in (2.46) needs to be made small. With the assistance of stability analysis, τ_n can be specified as

$$\tau_n = -k_\omega J_0 \omega_e + \begin{bmatrix} J_0 & 0 \end{bmatrix} \dot{\mu}_d + \begin{bmatrix} \delta_2 & 0 \end{bmatrix} r, \quad (2.47)$$

where $k_\omega \in \mathbf{R}^+$ is a control gain. The term $\dot{\mu}_d$, which is the derivative of μ_d , is evaluated as

$$\begin{aligned} \dot{\mu}_d &= B_\mu^{-1} [S(\omega) R^\top (\ddot{p}_{vgr} + g_v) + R^\top \ddot{p}_{vgr} - \alpha \dot{e}_v - \dot{e}_p - k_r \dot{r}] \\ &= B_\mu^{-1} [S(\omega) R^\top (\ddot{p}_{vgr} + g_v) + R^\top \ddot{p}_{vgr} - (k_r + \alpha I_2) \dot{r} + (\alpha^2 - 1) \dot{e}_p]. \end{aligned} \quad (2.48)$$

The closed-loop dynamics of r and ω_e can be evaluated by substituting the control input specified in (2.44) and (2.47) into the original open-loop dynamics equations. Substituting $\mu = \mu_d + \mu_e$ into (2.40) yields the closed-loop dynamics of r

$$\dot{r} = -k_r r + S(\omega) r + B_\mu \mu_e - e_p. \quad (2.49)$$

The closed-loop dynamics of ω_e is found by substituting (2.47) into (2.46) to yield

$$J_0 \dot{\omega}_e = -k_\omega J_0 \omega_e + \begin{bmatrix} \delta_2 & 0 \end{bmatrix} r. \quad (2.50)$$

2.4.2 End-effector Orientation Control

An error system of θ_{01} is formulated for the design of the end-effector orientation controller. The end-effector orientation tracking error, $e_{01}(t)$, is defined as the difference between the actual and desired orientation

$$e_{01} = \theta_{01} - \theta_{01r}, \quad (2.51)$$

and the time derivative is

$$\dot{e}_{01} = \dot{\theta}_{01} - \dot{\theta}_{01r}. \quad (2.52)$$

A filtered orientation tracking error signal, $r_2(t)$, can be expressed as

$$r_2 = \dot{e}_{01} + \beta e_{01}, \quad (2.53)$$

where $\beta \in \mathbf{R}^+$ is the control gain for end-effector orientation control. Dynamics of the filtered orientation tracking error is derived by taking time derivative of (2.53), multiplying both sides of the resulting equation with \bar{J}_1 (defined with (2.25)), and then substituting in θ_{01} dynamics from (2.25) to produce

$$\begin{aligned} \bar{J}_1 \dot{r}_2 &= \bar{J}_1 \ddot{\theta}_{01} - \bar{J}_1 \ddot{\theta}_{01r} + \bar{J}_1 \beta \dot{e}_{01} \\ &= \tau_1 - \xi_1 F - \bar{J}_1 \left(\ddot{\theta}_{01r} - \beta \dot{e}_{01} \right). \end{aligned} \quad (2.54)$$

Based on (2.54), the torque input to the manipulator joint, τ_1 , is designed as

$$\tau_1 = -k_2 \bar{J}_1 r_2 + \xi_1 F + \bar{J}_1 \left(\ddot{\theta}_{01r} - \beta \dot{e}_{01} - e_{01} \right) \quad (2.55)$$

in order to regulate the r_2 dynamics, where $k_2 \in \mathbf{R}^+$ is a control gain. External torque input on the VTOL aircraft can be calculated using

$$\tau_0 = \tau_n + \tau_1. \quad (2.56)$$

The closed-loop dynamics of r_2 becomes

$$\bar{J}_1 \dot{r}_2 = -k_2 \bar{J}_1 r_2 - \bar{J}_1 e_{01}, \quad (2.57)$$

which is equivalent to

$$\dot{r}_2 = -k_2 r_2 - e_{01} \quad (2.58)$$

as \bar{J}_1 is a scalar.

2.4.3 Stability Analysis

Lyapunov stability analysis is performed on the closed-loop error systems resulting from the proposed controller described in Sec. 2.4.1 and 2.4.2. A positive definite function is designed as

$$V = \frac{1}{2} e_p^\top e_p + \frac{1}{2} r^\top r + \frac{1}{2} J_0 \omega_e^2 + \frac{1}{2} r_2^2 + \frac{1}{2} e_{01}^2 \quad (2.59)$$

which has a time derivative

$$\dot{V} = e_p^\top \dot{e}_p + r^\top \dot{r} + J_0 \omega_e \dot{\omega}_e + r_2 \dot{r}_2 + e_{01} \dot{e}_{01}. \quad (2.60)$$

After substituting in \dot{e}_p from (2.34) and the closed-loop error dynamics of r , ω_e , r_2 expressed in (2.49), (2.50) and (2.58) respectively, it is found that

$$\begin{aligned} \dot{V} = & e_p^\top [S(\omega)e_p + e_v] + r^\top [-k_r r + S(\omega)r + B_\mu \mu_e - e_p] - k_\omega J_0 \omega_e^2 \\ & + \omega_e [\delta_2, 0]r - k_2 r_2^2 - r_2 e_{01} + e_{01} (r_2 - \beta e_{01}). \end{aligned} \quad (2.61)$$

As $S(\omega)$ is skew-symmetric, $\xi^\top S(\omega)\xi = 0$ for any $\xi \in \mathbf{R}^2$ and thus terms $S(\omega)e_p$ in the first bracket and $S(\omega)r$ in the second bracket will vanish. Substituting $r^\top e_p$ for $e_p^\top r$ and reorganizing terms yields

$$\dot{V} = e_p^\top (e_v - r) + r^\top B_\mu \mu_e - r^\top k_r r - k_\omega J_0 \omega_e^2 + \omega_e [\delta_2, 0]r - k_2 r_2^2 - \beta e_{01}^2. \quad (2.62)$$

By the definition in (2.41), B_μ is a diagonal matrix and $B_\mu = B_\mu^\top$; thus, $r^\top B_\mu \mu_e$ is rewritten as

$$r^\top B_\mu \mu_e = \mu_e^\top B_\mu r = -\omega_e [\delta_2, 0]r, \quad (2.63)$$

utilizing the definition of μ_e . Applying this identity to \dot{V} yields

$$\dot{V} = -\alpha e_p^\top e_p - r^\top k_r r - k_\omega J_0 \omega_e^2 - k_2 r_2^2 - \beta e_{01}^2 - e_p^\top \delta. \quad (2.64)$$

The first five terms are all less than zero. The upper-bound of last term can be found as

$$\begin{aligned} -e_p^\top \delta & \leq \|e_p\| \cdot \|\delta\| \\ & \leq \frac{1}{2} \left(\lambda_1 \|e_p\|^2 + \frac{1}{\lambda_1} \delta_2^2 \right), \end{aligned} \quad (2.65)$$

where $\lambda_1 \in \mathbf{R}^+$. Thus,

$$\dot{V} \leq - \left(\alpha - \frac{\lambda_1}{2} \right) e_p^\top e_p - r^\top k_r r - k_\omega J_0 \omega_e^2 - k_2 r_2^2 - \beta e_{01}^2 + \frac{1}{2\lambda_1} \delta_2^2. \quad (2.66)$$

A greater upper bound of \dot{V} can be written as

$$\begin{aligned} \dot{V} &\leq -\lambda_2 (\|e_p\|^2 + \|r\|^2 + J_0 \omega_e^2 + r_2^2 + e_{01}^2) + \frac{1}{2\lambda_1} \delta_2^2 \\ &\leq -2\lambda_2 V + \frac{1}{2\lambda_1} \delta_2^2, \end{aligned} \quad (2.67)$$

where a constant scalar $\lambda_2 \in \mathbf{R}^+$ is given by

$$\lambda_2 = \min \left\{ \left(\alpha - \frac{\lambda_1}{2} \right), k_{r1}, k_{r2}, k_\omega, k_2, \beta \right\}. \quad (2.68)$$

Solving the differential inequality in (2.67) yields

$$V \leq V_0 \cdot e^{-2\lambda_2 t} + \frac{1}{4\lambda_2 \lambda_1} \delta_2^2 (1 - e^{-2\lambda_2 t}), \quad (2.69)$$

where V_0 is V evaluated at $t = 0$.

A complete tracking error vector of the control system is defined as

$$\eta = \left[e_p^\top, r^\top, \sqrt{J_0} \omega_e, r_2, e_{01} \right]^\top, \quad (2.70)$$

which can be rewritten as

$$\frac{1}{2} \|\eta\|^2 = V. \quad (2.71)$$

Substituting (2.71) into (2.69), multiplying by 2 and taking square root on both sides yields,

$$\|\eta(t)\| \leq \sqrt{\|\eta_0\|^2 e^{-2\lambda_2 t} + \frac{1}{2\lambda_2 \lambda_1} \delta_2^2 (1 - e^{-2\lambda_2 t})}, \quad (2.72)$$

where η_0 is η evaluated at $t = 0$. This result represents the steady state bound on $\|\eta(t)\|$ as

$$\lim_{t \rightarrow \infty} \|\eta(t)\| = \frac{\delta_2}{\sqrt{2\lambda_1\lambda_2}}. \quad (2.73)$$

In other words, the norm of the complete tracking error, $\eta(t)$, is globally uniformly ultimately bounded (GUUB).

Further analysis can be done to show all signals are bounded. By (2.71), V is bounded. From (2.73) and the definitions of η in (2.70), e_p , r , ω_e , r_2 , and e_{01} are bounded. By definition of r in (2.37), e_v is bounded. Since $R \in SO(2)$, and $\|R\| \equiv 1$, p_{vg} and $\dot{p}_{vg} \in \mathcal{L}_\infty$ by e_p and e_v definitions and the smoothness assumption of the reference trajectories. From (2.42), we can observe μ_d is bounded, which also means ω_d and $F \in \mathcal{L}_\infty$. Further, we can show ω is bounded by definition of ω_e . From definition of μ_e , it is bounded because ω_e is bounded and $F_e = 0$. Signals \dot{r} and \dot{e}_p are bounded by the closed-loop dynamics of r in (2.49) and (2.34), respectively, which leads to $\dot{\mu}_d \in \mathcal{L}_\infty$. From (2.47), the net torque τ_n is bounded. From definition of r_2 in (2.53), $\dot{e}_{01} \in \mathcal{L}_\infty$ because r_2 and e_{01} are bounded. Thus, we know τ_1 is bounded from the control law defined in (2.55) and τ_0 is bounded from (2.56). Therefore, we conclude that all signals are bounded in the closed-loop system.

2.5 Experimental Results

Experiments are implemented in order to evaluate the performance of the controller. Numerical simulation of the planar system is constructed with the validated dynamics model and proposed controller. A circular reference trajectory is employed in the simulation and both position and orientation trajectory tracking error of the end-effector is shown. The physical test-bed is constructed by Ran Huang. It adopted

a pan-tilt mechanism with relatively large radius to mimic the constraint of a two dimensional space, which is difficult to reproduce in real world without significant interaction with the interested dynamics of MOVA system. Design and construction of the test-bed is briefly discussed in this section and the implementation details are covered in work of Huang [37]. Experimental results on the physical test-bed demonstrate clearly that for the MOVA system, a unified controller offers superior system performance than controllers designed using the separate control strategy.

2.5.1 Simulation Results

Numerical simulations of the planar MOVA system with the proposed controller were performed with Simulink[®] to show that the proposed controller is able to achieve the goal of end-effector trajectory tracking with acceptable error. The simulated plant was previously validated in Sec. 2.3.3. The same planar MOVA system physical parameters (Table 2.2) are used for simulation as in Sec. 2.3.3 where the plant MOVA system dynamics is validated. Note that the mass and moment of inertia of the VTOL aircraft body and the one-link manipulator are comparable.

The reference trajectory of the end-effector (see Figure 2.10) is a 5m diameter circular trajectory centered at the origin, moving at angular rate $\omega = 0.5$ rad/s. The reference orientation of the end-effector follows a sine wave with the same ω . This trajectory is described by the following function

$$\begin{aligned} p_{er}(t) &= \begin{bmatrix} A \cos(\omega t) \\ A \sin(\omega t) \end{bmatrix}, \\ \theta_{01r}(t) &= \frac{1}{2}\pi \sin(\omega t), \end{aligned} \tag{2.74}$$

where $A = 5$ m and $\omega = 0.5$ rad/s. The simulation lasts enough time for the trajectory

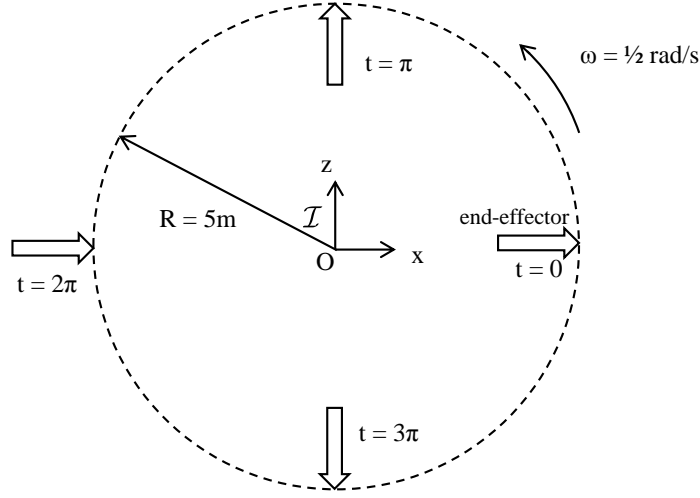


Figure 2.10: Trajectory of end-effector for simulation is shown as the dashed circle. The tip of arrow (not to scale) denotes the end-effector of the MOVA system.

to go around a full circle.

The initial conditions for the simulation are $p_0(0) = [4.5, 0]^T$ m, $\theta_0(0) = 0$ rad and $\theta_1(0) = 0$ rad, which results in $p_e(0) = p_{er}(0) = [5, 0]^T$ m. The initial velocity of both the aircraft and the onboard manipulator are zero. In other words, the MOVA system starts statically with the end-effect at the desired location. Controller gains are set to the following values for this simulation: $k_r = 10I_2$, $k_\omega = 10$, $\delta_2 = 0.02$, $\alpha = 10$, $\beta = 10$, $k_2 = 10$.

The results of the simulation are captured in Figures. 2.11 to 2.13. Fig. 2.11 shows the position error vector of the end-effector, projected onto the x- and z- axis of the inertial frame \mathcal{I} . The same error is also illustrated by plotting the norm of error in Fig. 2.12. These two figures show that the position error of the end-effector starts at zero and rapidly ramps up due to the mismatch between the initial velocity command and the condition that the entire system has zero velocity initially. Both of the error components and the norm of the error quickly decays after reaching a peak of 0.13 m to less than 0.005 m at around $t = 1.5$ sec and keeps below that

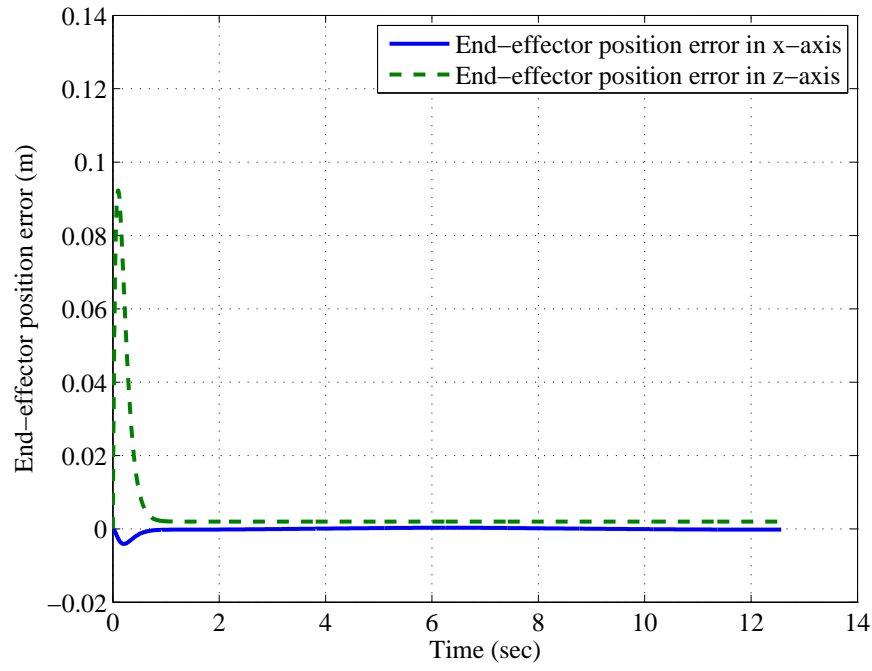


Figure 2.11: End-effector position error projected onto x- and z-axis of \mathcal{I}

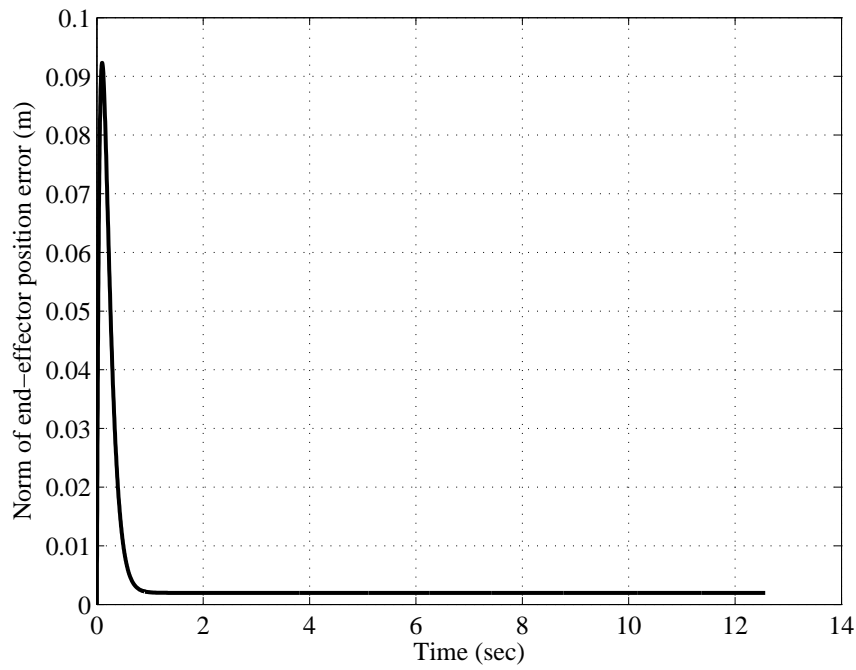


Figure 2.12: Norm of end-effector position error

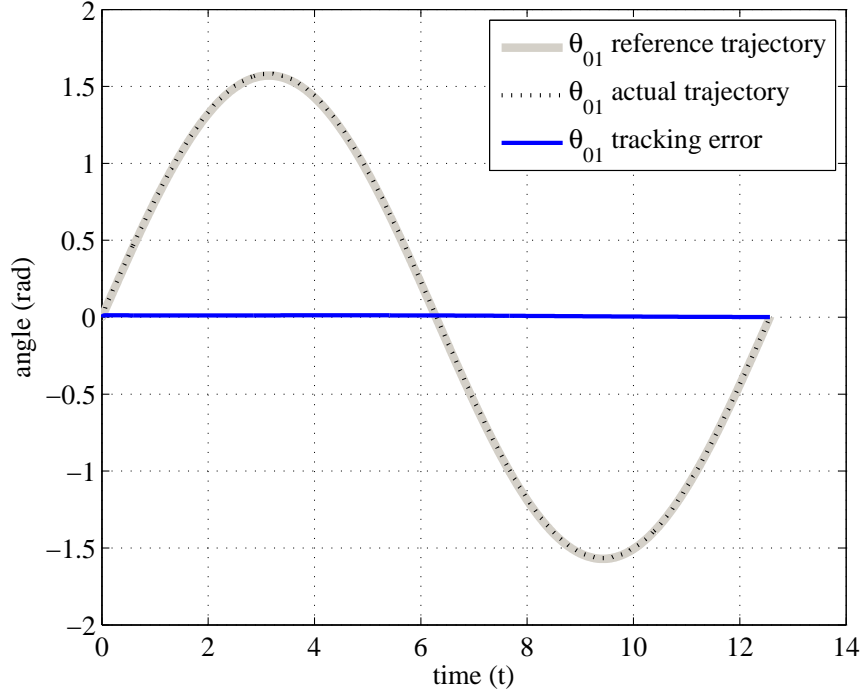


Figure 2.13: Plot of $\theta_{01r}(t)$, the reference trajectory of end-effector orientation, $\theta_{01}(t)$, the actual trajectory and $e_{01}(t)$, the tracking error.

level for the rest of simulation. Given the magnitude of the actual trajectory, the tracking error is relatively small percentage, disregarding the initial “catching up” period. The end-effector orientation tracking performance is shown in Fig. 2.13. The tracking error, $e_{01}(t)$, keeps closely to zero during the entire period of simulation and the actual orientation of end-effector almost overlaps with the reference trajectory, indicating a very good tracking result.

The simulation results shown are suggestive in that the tracking error of both the end-effector position and end-effector orientation match a small bounded error predicted by the theoretical result. Taken together, the modeling, control design, and simulation results indicate that the goal of a unified controller for the fully modeled planar MOVA system has been achieved.

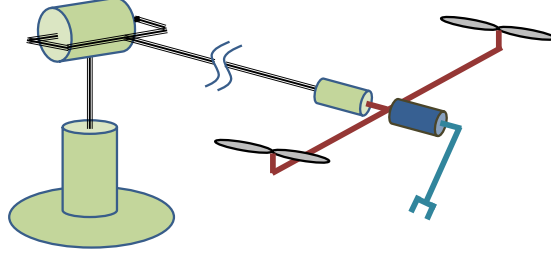


Figure 2.14: Illustration of the proposed hardware-in-the-loop test bed. The light colored cylinders represent passive joints of the test-bed, which also serve in measurement of VTOL position and orientation. The triple lines represents links of the test-bed while single lines illustrate frames and the manipulator link of the 2D MOVA system. The dark colored cylinder is the joint of the onboard manipulator.

2.5.2 Experimental Result from Physical Test-bed

A hardware-in-the-loop experimental test-bed of the planar MOVA was constructed to demonstrate the performance of the controller under realistic constraint of implementation. The test-bed was designed in collaboration with Ran Huang and the plots along with some system description are shared and appear both here and in Huang’s thesis [37]. Considering the difficulty in physically restraining a system to move only in a 2D plane, as the planar MOVA system assumed, without affecting the dynamics of the system being tested, a spherical approximation is made. Instead of being constrained in a vertical plane, the 2D VTOL aircraft of planar MOVA is attached to a long, light-weight passive manipulator arm which is attached to the inertial frame via a spherical joint (Fig. 2.14). The resulting system has the same number of degrees-of-freedom as the planar MOVA system.

In Fig. 2.15, a photo of the test-bed implementation is shown. The passive manipulator is formed by a pan-tilt mechanism and a revolute joint near the VTOL aircraft. Optical incremental encoders are used for VTOL aircraft position and attitude feedback. The VTOL aircraft is realized by a “twin-rotor” aircraft, which has

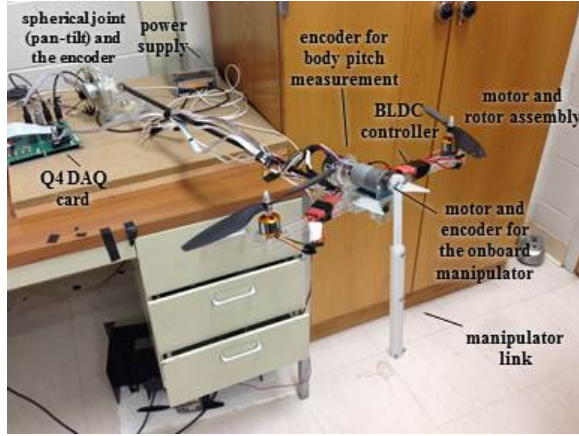


Figure 2.15: Photo of the planar MOVA hardware-in-the-loop test-bed.

Table 2.3: Physical Parameters of Planar MOVA test-bed

Parameter	Value	Parameter	Value
m_0	0.780 kg	J_0	$5.86 \times 10^{-3} \text{ kg}\cdot\text{m}^2$
m_1	0.059 kg	J_1	$4.14 \times 10^{-5} \text{ kg}\cdot\text{m}^2$
l_1	0.3 m	l_p	0.95 m

two rotors to provide the lift force and torque on the aircraft body. The onboard manipulator is driven by a DC motor with optical encoder position feedback. Physical parameters of testbed components are listed in Table. 2.3. The parameter l_p is the length of passive manipulator arm of the test bed. Mass of the passive manipulator arm is added into the m_0 . Values of the variables x_0 and z_0 are obtained using the spherical approximation by multiply l_p to corresponding passive manipulator joint angle for the experiment.

In order to highlight the rationale for the proposed controller, an experiment to compare between a separate control strategy and the proposed unified controller was performed. The separate controller controls the “twinrotor” and onboard manipulator individually without considering dynamics coupling. The onboard manipulator arm is controlled using a PD (Proportional-derivative) controller. To ensure the validity of the comparison, the separate “twinrotor” controller is also derived using back-

stepping method following [4]. The same values are used for shared gains between experiment.

Desired position and attitude trajectory of the end-effector for this experiment are described by

$$\begin{aligned}\theta_{01r}(t) &= 0.8 \sin(1.5t) - \pi/2, \\ p_{er}(t) &= \begin{bmatrix} l_1 \cos \theta_{01r}(t) \\ l_1 \sin \theta_{01r}(t) \end{bmatrix}.\end{aligned}\tag{2.75}$$

From the kinematics equation (2.2), it is found that the desired position for the CM of the VTOL aircraft stays at origin and the desired manipulator movement is seemingly the trajectory of a pendulum.

The controller gains of the unified MOVA controller used in this experiment are: $k_r = 0.08I_2$, $k_\omega = 0.03/J_0$, $\alpha = 0.05$, $\delta_2 = 0.25$, $\beta = 15$, and $k_2 = 100$. The proportional and derivative gains of the onboard manipulator in the separate controller are 15 and 1, respectively.

The test results are demonstrated in two figures comparing outcome from the unified controller and the separate controller. In Fig. 2.16, the norm of end-effector position error is plotted over time, which shows the general performance of the two controllers. During experiment, the system was hand launched, the initial condition cannot be guaranteed to be the same for both tests. The data displayed was measured at steady state starting from $t = 25$ sec. For the unified controller, the error is around 0.1 m during the entire period and the fluctuation is less than 0.05 m. On the other hand, the error of the separate controller has average value at about 0.18 m and maximum value more than 0.25 m. Moreover, the error from the separate controller clearly shows sine wave pattern which may corresponds to

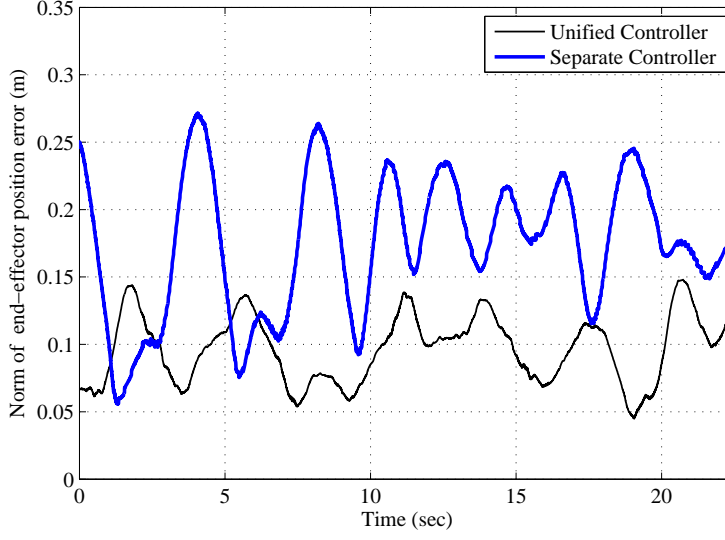


Figure 2.16: Comparison of norm of end-effector position error at steady state from $t = 25$ sec to $t = 45$ sec

dynamics interaction between the onboard manipulator and the VTOL aircraft as the desired trajectory of the end-effector.

The other plot (Fig. 2.17) shows dynamical disturbance of the onboard manipulator on the position of the VTOL aircraft more obviously. The horizontal position, x_0 , is plotted since it is more susceptible than z_0 under the disturbance of onboard manipulator when the end-effector has a desired trajectory described in (2.75). Kinematics analysis shows that x_0 is supposed to stay at 0 m following the desired trajectory. It is clear that the result from the separate controller has greater fluctuations than that from the unified controller. The horizontal location of the VTOL aircraft varies from -0.07 m to +0.04 m using the unified controller, while the same variable oscillate between -0.2 m and 0.13 m. Like Fig. 2.16, the sinusoidal x_0 of separate controller result demonstrates the effect of the unmatched dynamics during controller design in a close-loop system. There are approximately 5.5 cycles in the sinusoidal curve of x_0 in the 22.5 seconds range in the separate controller result, which matches

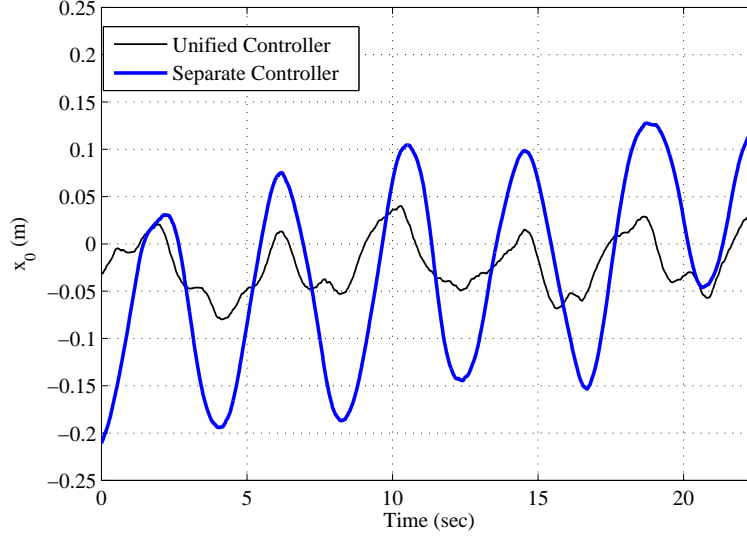


Figure 2.17: Comparison of VTOL aircraft coordinate x_0 . The reference trajectory of end-effector attitude is added to show the direct effect of dynamics disturbance of the onboard manipulator on the VTOL aircraft.

closely ($2\pi \times 5.5/22.5 = 1.535$) to the frequency of desired trajectory of the onboard manipulator.

Both plots demonstrates that the proposed unified controller shows better performance than the control strategy that controls the VTOL aircraft and the onboard manipulator separately. The advantage of unified controller is attribute to the proper handling of dynamics coupling between the manipulator and VTOL aircraft.

2.6 Conclusion

A planar MOVA system construction with 2D VTOL aircraft and a single link onboard manipulator was illustrated. Dynamics equations of this planar MOVA system, derived by the Lagrangian method, suggest that the control strategy should compensate for the coupling between the manipulator and the VTOL aircraft. The equations are transformed into a form with translational and rotational dynamics

decoupled using the alternative kinematics derived by virtual manipulator method. A unified MOVA controller is designed based on the decoupled dynamics equations and proven to have GUUB tracking performance via Lyapunov type stability analysis. Simulation of the proposed controller is performed and the result is satisfying in terms of the end-effector tracking error. Construction of a physical test-bed of the planar MOVA system is described and experimental results on the test-bed was shown. Comparison of results from the proposed controller and a controller designed using a separate strategy demonstrate that the proposed controller has advantage of compensating for subsystem coupling. The approach adopted for planar MOVA dynamics modeling and control system design in this chapter is able to be generalized and applied to the full three dimensional MOVA system.

Chapter 3

Systematic Approach to MOVA

System Dynamics Derivation

3.1 Introduction

The aerial nature gifted MOVA systems with flexibility and maneuverability, but also suggests difficulties from a control perspective owing to its complex dynamics. Categorized as a multi-body system, the dynamics of a MOVA system is inherently intricate due to its high degree-of-freedom configuration space and interaction forces and torques between bodies. This calls for a model-based control design approach and a unified control strategy that accounts for interaction between body and the onboard manipulator. An accurate dynamics model is the prerequisite for model-based control design and is useful in constructing simulations for evaluation of the proposed controller. Previous planar case study gives a successful example for dynamics derivation and controller design. However, 3D MOVA system dynamics derivation is expected to be more complex than the planar case as the number of DOF of the system increases. Manual derivation of dynamics equations does not scale up

a practical 3D MOVA system with multiple number of joints. Thus, it is necessary to develop a systematic framework for dynamics derivation and implementation of such framework so the equations of motion of MOVA system can be automatically generated.

3.1.1 Previous Work

Derivation of the MOVA system dynamics model is closely related to that of other multi-body dynamics systems, such as satellite manipulators and underwater manipulation systems. Satellite based manipulator research communities were the first to investigate mobile manipulator dynamics by augmenting the general dynamics derivation with a framework that helps with description of the Lagrangian energy equation and yields clearer results in a more compact format [15]. Researchers of Unmanned Under-water Vehicle (UUV) based manipulators utilized a similar approach for dynamics modeling, yet with concentration on resolving issues caused by hydrodynamics interaction on the system [18, 19, 20]. The relatively high density of water inevitably induces hydrodynamics terms that cannot be ignored.

A generalized framework for derivation of the dynamics equations of multi-body systems is detailed in Wittenburg's book, in which commercial software that adopts this formalism are introduced [38]. Besides the software mentioned in this book, other automatic dynamics derivation programs exist. SPACEMAPLE is a program that derives dynamics equations for satellite manipulators, and was developed using the algorithms proposed by Moossavian and Papadoupoulos [17]. Neweul-M² is a general multi-body system dynamics derivation and simulation software developed by Kurz and colleagues [39, 40]. It is able to output closed-form dynamics equations by using the symbolic engine in MATLAB for the derivation process. Commercial

dynamics derivation software described in [38] cannot be used for the MOVA dynamics investigation due to availability issues. SPACEMAPLE is developed for space manipulator and does not include effects of gravity and aerodynamics. Neweul-M² is the only software we are able to obtain for multi-body system dynamics derivation via individual requests. However, it requires manual specification of system geometry and mass properties using a command line interface, making it difficult to use given it lacks the interface with other 3D modeling software for parameters extraction. Moreover, it is not available to the general public such as researchers in universities or hobbyists.

3.1.2 Contribution

A systematic approach for deriving the dynamics equations of the MOVA system with a single onboard manipulator composed of revolute joints is developed here. In Sec. 3.2, bary-center representation is used in the kinematics development as previous research on satellite-based system. Dynamics equations of MOVA system are then derived using the Lagrangian approach in Sec. 3.3. The entire derivation process uses matrix algebra uniformly to ensure the conciseness. Explicit and general dynamics equation for MOVA system is presented in closed-form as the result. A slightly different kinematics and dynamics development is adopted in this chapter compared to that for the planar system described in Chapter 2 in order to ensure a more concise derivation. For the planar system, the dynamics derivation was performed using trivially defined kinematics and then an alternative kinematics representation obtained from virtual manipulator approach was substituted into the dynamics equation to yield a decoupled form. Here the linear velocity expressions of individual rigid bodies are constructed with the linear velocity of the whole system center-of-mass

as a component so that later in the dynamics derivation the outcome is the in the decoupled form directly. A program (named MOVADYN) is developed in Sec. 3.4 to automatically perform the derivation with given system parameters that specify mass and dimensional properties. This program is compatible with parameters given in symbolic form as it is written to run in a symbolic engine. Additional accessory programs are also developed to export parameter information from a 3D modeling software and to convert the derived dynamics equation into code and other files that can be directly incorporated simulation or controller implementation, saving time from tedious manual coding process. The MOVADYN program and accessories were released as an open-source project, made available to both researchers in academia and hobbyists in order to boost advances in this field. In Sec. 3.5, validation of the MOVADYN program is performed by comparing the automatically generated output with hand-derived result and comparing the simulation results of the derived dynamics equation with that from numerical rigid-body dynamics simulator. Both methods show that the MOVADYN program outputs correct dynamics equations for the specified system.

3.2 System Description and Kinematics

An abstract definition of the MOVA system is presented and the kinematics expressions of linear and angular velocity are developed in this section to facilitate later dynamics equation derivation based on a Lagrangian approach.

3.2.1 System

A general MOVA system consists of a VTOL aircraft of any type and one or more onboard manipulators. In this work, various types of VTOL aircraft are

abstracted as one free-floating rigid body with a body-fixed thrust vector and torque inputs in three orthogonal axes (shown in Fig. 1.3). The onboard device is assumed to be a single serial manipulator constructed of rigid links connected by revolute joints. One end of the manipulator, the base, is attached to the VTOL body while the other is considered as the end-effector.

The links are numbered Link 1 to N , starting from Link 1 closest to the VTOL aircraft body increasing to Link N at the end-effector. Since the VTOL aircraft and manipulator links are all rigid bodies, they are uniformly numbered Body 0 to N as well, where the VTOL aircraft is denoted as Body 0 and the numbering proceeds to the end-effector, Body N .

3.2.2 Definitions

A few definitions are made to facilitate the MOVA system kinematics derivation. A reference frame is affixed to the rigid aircraft body and each manipulator link and are referred to as \mathcal{B}_i for $i = 0$ to N for Body 0 to N , respectively. The inertial frame is named \mathcal{I} . The origin of the inertial frame is point O . Center of mass (CM) of the entire MOVA system is denoted as point C in the inertial frame.

The symbol p is uniformly used for naming linear position vectors in 3D space. Decorations are used to differentiate individual vectors, the reference point and the frame of representation. A subscript A/B denotes the start, A , and the end, B , of a vector. Besides point C and origin O , A and B can take integer values $i \in [0, N]$, denoting a start or end point on the center of mass (CM) of Body i . Short-hand notation $p_{A/O} \triangleq p_A$ is defined for concise representation of frequently used vectors. Left superscripts denote the frame of representation: I denotes the inertial frame \mathcal{I} and integer $i \in [0, N]$ denotes body fixed frame \mathcal{B}_i . Left superscript 0, which denotes

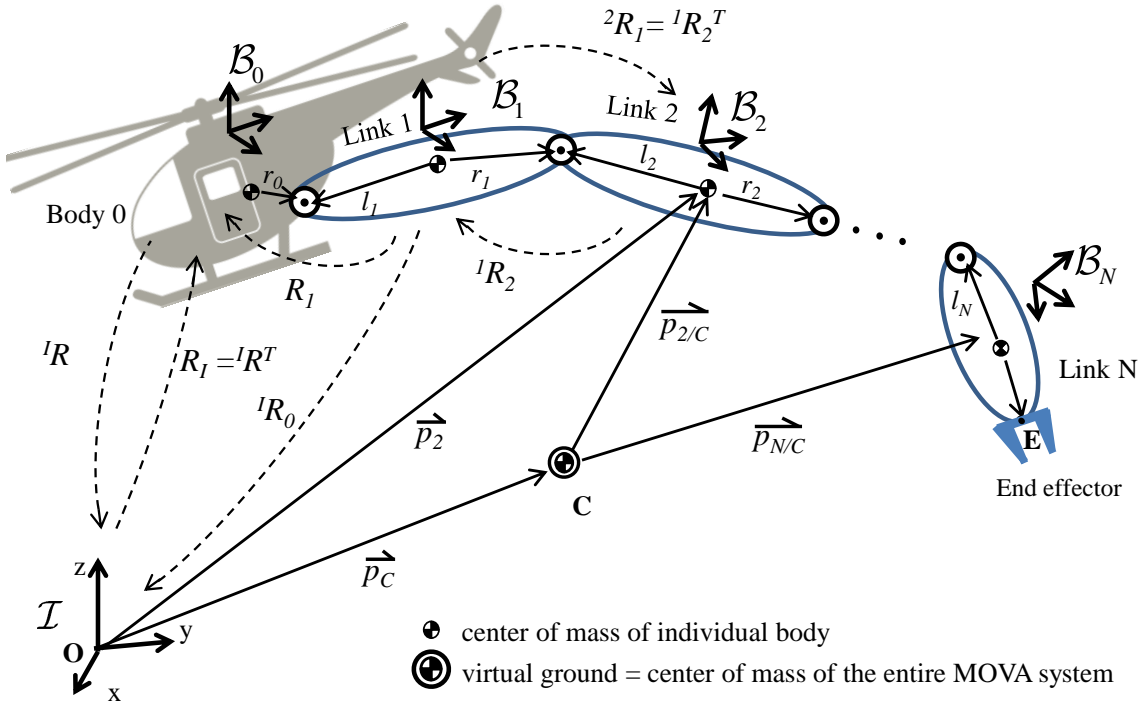


Figure 3.1: Illustration of a general MOVA system frame definition and notations conventions

Table 3.1: List of notation

Symbol	Meaning
\mathcal{I}	The inertial frame
$\mathcal{B}_i, i \in [0, N]$	Frame attached to body i
${}^I p_{A/B}$	Vector from point A to B in frame \mathcal{I} .
${}^i p_{A/B}, i \in [0, N]$	Vector from point A to B vector in frame \mathcal{B}_i
$p_{A/B}$	Vector from point A to B in frame \mathcal{B}_0
${}^I p_A$	Vector from origin to A in frame \mathcal{I}
${}^i p_A, i \in [0, N]$	Vector from origin to A in frame \mathcal{B}_i
p_A	Vector from origin to A in frame \mathcal{B}_0
${}^I p_{i/A}, i \in [0, N]$	Vector from center of mass of Body i to point A in frame \mathcal{I}
${}^i p_{i/A}, i \in [0, N]$	Vector from center of mass of Body i to point A in frame \mathcal{B}_i
$p_{i/A}, i \in [0, N]$	Vector from center of mass of Body i to point A in frame \mathcal{B}_0
${}^I \omega_{i/j}, i, j \in [0, N]$	Angular velocity of \mathcal{B}_i using \mathcal{B}_j as reference in frame \mathcal{I}
${}^k \omega_{i/j}, i, j, k \in [0, N]$	Angular velocity of \mathcal{B}_i using \mathcal{B}_j as reference in frame \mathcal{B}_i
$\omega_{i/j}, i, j \in [0, N]$	Angular velocity of \mathcal{B}_i using \mathcal{B}_j as reference in frame \mathcal{B}_0
${}^I \omega_i, i \in [0, N]$	Angular velocity of \mathcal{B}_i using \mathcal{B}_0 as reference in frame \mathcal{I}
${}^k \omega_i, i, k \in [0, N]$	Angular velocity of \mathcal{B}_i using \mathcal{B}_0 as reference in frame \mathcal{B}_i
$\omega_i, i \in [0, N]$	Angular velocity of \mathcal{B}_i using \mathcal{B}_0 as reference in frame \mathcal{B}_0
ω	Angular velocity of \mathcal{B}_0 using \mathcal{I} as reference in frame \mathcal{B}_0
${}^j R_i, i, j \in [0, N]$	Rotation matrix from \mathcal{B}_i to \mathcal{B}_j
$R_i, i \in [0, N]$	Rotation matrix from \mathcal{B}_i to \mathcal{B}_0
${}^i R, i \in [0, N]$	Rotation matrix from \mathcal{B}_0 to \mathcal{B}_i
$R_I, i \in [0, N]$	Rotation matrix from \mathcal{I} to \mathcal{B}_0
${}^I R, i \in [0, N]$	Rotation matrix from \mathcal{B}_0 to \mathcal{I}

the body frame (\mathcal{B}_0) of the VTOL aircraft, is the default frame and can be omitted.

The abbreviation format to show the frame of representation is the same as that defined above for vectors. The symbol ω is uniformly used for the angular velocities of rigid bodies. Angular velocity of frame i with respect to frame j is written as ${}^I \omega_{i/j}$, where $i, j \in [0, N]$ refers to frame \mathcal{B}_i and \mathcal{B}_j and I denotes inertial frame \mathcal{I} . Similar to the point notation, defaults are defined for concise use: i in can be omitted if it is 0 and the reference body in the subscript is left out if the inertial frame is used. This way, ω_i denotes absolute angular velocity of Body i represented in frame 0. A list of notation is provided in Table 3.1 for reference.

3.2.3 Pose and Angular Velocity of Rigid Bodies

Transformation of a vector from frame \mathcal{B}_i to \mathcal{B}_j is described by the rotation matrix, ${}^jR_i \in SO(3)$. In other words, for a vector $x \in \mathbf{R}^3$, it is easy to change the frame of representation from \mathcal{B}_i to \mathcal{B}_j using

$${}^jx = {}^jR_i {}^ix. \quad (3.1)$$

A left superscript I , on the rotation matrix, IR_i , denotes transformation of a vector from \mathcal{B}_i to the inertial frame. Transformation from or to \mathcal{B}_0 is frequently used and thus \mathcal{B}_0 is the default frame if a super or subscript is omitted.

Transformation from any body frame \mathcal{B}_i to the inertial frame can be evaluated as

$${}^IR_i = {}^IR_0 {}^0R_i = {}^IRR_i, \quad (3.2)$$

where, $R_0 = I_3$ and

$$R_i = {}^0R_1 {}^1R_2 \dots {}^{i-1}R_i. \quad (3.3)$$

For a rigid body i in the multi-body system, its angular velocity with respect to the inertial frame \mathcal{I} can be found in a chained form

$${}^I\omega_i = {}^I\omega + \sum_{j=1}^i {}^IR_{j-1} {}^{j-1}\omega_j, \quad (3.4)$$

since the instantaneous rotation of a series of objects can be summed linearly. Representation of the same relationship in \mathcal{B}_0 is obtained by pre-multiplying the rotation matrix ${}^0R_I = {}^IR_0^T$ to yield

$$\omega_i = \omega + \sum_{j=1}^i R_{j-1} {}^{j-1}\omega_j, \quad (3.5)$$

where the frame of reference \mathcal{B}_0 is the default and not explicitly marked at the left superscript position and ω is the aircraft body angular velocity in frame \mathcal{B}_0 .

The variable ${}^{i-1}\omega_i$ is the angular velocity of Body i , which is also Link i of the onboard manipulator, with respect to the previous link toward the VTOL aircraft, Link $i - 1$. Since all joints are revolute, it is determined by the joint velocity and the rotation axis of Joint i . Thus, there exist

$${}^{i-1}\omega_i = u_i \dot{q}_i, \quad (3.6)$$

where \dot{q}_i is the joint velocity of Joint i and $u_i \triangleq {}^{i-1}u_i$ denotes the rotation axis of Joint i represented in frame \mathcal{B}_{i-1} . This notation then defines q to be the joint configuration vector of the onboard manipulator, $q = [q_1, q_2, \dots, q_N]^T$ where q_i is the joint position of a specific link, and \dot{q} is thus the vector of joint velocities. The angular velocity of Body i in \mathcal{B}_0 can be written in matrix representation

$$\omega_i = \omega + F_i \dot{q}, \quad (3.7)$$

where, F_i is a Jacobian matrix for angular velocity, which can be written as concatenation of a series of vectors

$$F_i = \begin{bmatrix} R_0 u_1 & R_1 u_2 & \dots & R_{i-1} u_i & O_{3 \times (N-i)} \end{bmatrix}, \quad (3.8)$$

and $O_{m \times n}$ denotes a zero matrix with dimension specified by the subscript.

3.2.4 Position and Velocity of an Arbitrary Point

Position of the CM of each rigid body of the MOVA system, p_i , is needed to obtain the corresponding velocity for calculation of total kinetic energy. In the inertial frame, there is

$${}^I p_i = {}^I p_C + {}^I p_{i/C}. \quad (3.9)$$

The frame of representation of position vectors can be changed from a body frame, e.g. $p_{i/C}$, by left multiplying the appropriate rotation matrix and yields

$${}^I p_i = {}^I p_C + {}^I R p_{i/C}. \quad (3.10)$$

Due to the serial configuration of the manipulator,

$$p_{i/C} - p_{i-1/C} = r_{i-1} - l_i, \quad (3.11)$$

where l_i and r_i are vectors from CM of Body i to the previous joint (Joint $i - 1$) and the next joint (Joint i) on the manipulator (see Fig. 3.1), respectively. It is worth noting that ${}^i r_i$ and ${}^i l_i$ are both constant because all links and the VTOL aircraft are rigid. Vector $l_0 = 0$ and $r_N = 0$ as there is no joint prior to Body 0 and after Body N.

Also, the relation

$$\sum_{i=0}^N m_i p_{i/C} = 0 \quad (3.12)$$

holds since point C is the center of mass of the system, where m_i is the mass of Body i . Solving the vector equation set formed by (3.11) for integer $i \in [1, N]$ and (3.12), there is

$$p_{i/C} = \sum_{j=1}^i (r_{j-1} - l_j) \mu_i - \sum_{j=i+1}^N (r_{i-1} - l_i) (1 - \mu_i), \quad (3.13)$$

where $\mu_i = \sum_{j=0}^{i-1} m_j / M_t$ and the total mass $M_t = \sum_{i=0}^N m_i$.

Regrouping r and l vectors that bear the same index together, results in

$$p_{i/C} = \sum_{j=0}^N v_{ij}, \quad (3.14)$$

where

$$v_{ij} = r_j(\mu_{j+1} - \mathbf{1}_{j-i}) - l_j(\mu_j - \mathbf{1}_{j-i-1}), \quad (3.15)$$

and $\mathbf{1}_i$ is the discrete step function and $\mathbf{1}_i = 1$ if $i \geq 0$ and $\mathbf{1}_i = 0$ otherwise. From the fact that ${}^i r_i$ in which ${}^i l_i$ are constant vectors,

$${}^j v_{ij} = {}^j r_i(\mu_{j+1} - \mathbf{1}_{j-i}) - {}^j l_j(\mu_j - \mathbf{1}_{j-i-1}) \quad (3.16)$$

is a constant vector as well. It is convenient to separate the part of (3.16) that changes with the configuration variable of the manipulator q and rewrite p_i as

$$p_{i/C} = \sum_{j=0}^N R_j^j v_{ij}. \quad (3.17)$$

The velocity of a certain point is found by taking the time derivative of the position vector. However, the derivative of the coordinates of a position vector equals to the velocity of the point only if the position vector is represented in an inertial frame. Dot notations, i.e. a dot over the variable, are dedicated to denote time derivative of coordinate points and vectors represented in the inertial frame. According to this convention, ${}^I \dot{p}_C = \frac{d}{dt}({}^I p_C)$ is the velocity of the CM of the MOVA represented in the inertial frame and $\dot{p}_C = R_I \frac{d}{dt}({}^I p_C)$ is the velocity of the CM of the MOVA represented in the inertial frame \mathcal{B}_0 . Note that $\frac{d}{dt} p_C$ is the time derivative of the coordinates of

point C in frame \mathcal{B}_0 and $\dot{p}_C \neq \frac{d}{dt}p_C$, because

$$\begin{aligned}
\dot{p}_C &= R_I \frac{d}{dt}({}^I R p_C) \\
&= R_I \left[\left(\frac{d}{dt} {}^I R \right) p_C + {}^I R \frac{d}{dt} p_C \right] \\
&= R_I \left({}^I R S(\omega) p_C + {}^I R \frac{d}{dt} p_C \right) \\
&= S(\omega) p_C + \frac{d}{dt} p_C.
\end{aligned} \tag{3.18}$$

The velocity of the CM of Body i is thus

$${}^I \dot{p}_i = {}^I \dot{p}_C + {}^I \dot{p}_{i/C} \tag{3.19}$$

where

$${}^I \dot{p}_{i/C} = \frac{d}{dt}({}^I R p_{i/C}) = S({}^I \omega) {}^I R p_{i/C} + {}^I R \frac{d}{dt}(p_{i/C}). \tag{3.20}$$

In frame \mathcal{B}_0 , $\dot{p}_{i/C}$ is represented as

$$\begin{aligned}
\dot{p}_{i/C} &= R_I S({}^I \omega) {}^I R p_{i/C} + R_I {}^I R \frac{d}{dt}(p_{i/C}) \\
&= S(\omega) p_{i/C} + \frac{d}{dt}(p_{i/C}),
\end{aligned} \tag{3.21}$$

utilizing the fact that

$$R_I S({}^I \omega) {}^I R = {}^I R^\top S({}^I R \omega) {}^I R = S(\omega). \tag{3.22}$$

By (3.17), $\frac{d}{dt}p_{i/C}$ is evaluated by taking time derivative on each term, which yields

$$\frac{d}{dt}p_{i/C} = \sum_{j=0}^N \frac{d}{dt}(R_j^j v_{ij}) = \sum_{j=0}^N S(\omega_{j/0}) R_j^j v_{ij}. \quad (3.23)$$

Substituting (3.17) and (3.23) into (3.21), there is

$$\begin{aligned} \dot{p}_{i/C} &= S(\omega) \sum_{j=0}^N S(\omega) R_j^j v_{ij} + \sum_{j=0}^N S(\omega_{j/0}) R_j^j v_{ij} \\ &= \sum_{j=0}^N S(\omega + \omega_{j/0}) R_j^j v_{ij} \\ &= - \sum_{j=0}^N S(R_j^j v_{ij}) \omega_j, \end{aligned} \quad (3.24)$$

having used an identity of skew-symmetric matrix

$$S(a)b = -S(b)a \quad a, b \in \mathbf{R}^3. \quad (3.25)$$

Further, replacing ω_j from (3.7) yields

$$\dot{p}_{i/C} = - \sum_{j=0}^N S(R_j^j v_{ij}) \omega - \sum_{j=0}^N S(R_j^j v_{ij}) F_j \dot{q}. \quad (3.26)$$

Thus, the velocity of Body i with respect to, the original of the inertial frame, O , in inertial frame is

$$\dot{p}_i = \dot{p}_C + J_{\omega i} \omega + J_{qi} \dot{q}, \quad (3.27)$$

where

$$\begin{aligned} J_{\omega i} &= - \sum_{j=0}^N S(R_j^j v_{ij}), \\ J_{qi} &= - \sum_{j=0}^N S(R_j^j v_{ij}) F_j. \end{aligned} \tag{3.28}$$

Equations (3.7), (3.17) and (3.27) will be used for forming the expressions of kinetic energy and potential energy in the Lagrangian dynamics derivation.

3.3 Derivation of Dynamics Equation

3.3.1 Representation of the Lagrangian

The Lagrangian L is the difference between kinetic energy, T , and potential energy, V , *i.e.* $L=T-V$. The kinetic energy of the entire MOVA system is the summation of the kinetic energy of each body due to translation or rotation movement

$$T = \frac{1}{2} \sum_{i=0}^N m_i \dot{p}_i^\top \dot{p}_i + \frac{1}{2} \sum_{i=0}^N ({}^i R \omega_i)^\top J_i ({}^i R \omega_i) = T_1 + T_2, \tag{3.29}$$

where J_i is the moment of inertia matrix of Body i along the direction defined by \mathcal{B}_i , T_1 and T_2 are summations of kinetic energy resulting from translation and rotation of each body, respectively. Potential energy is evaluated directly from the position of point C as

$$V = -M_t g_v^\top p_C, \tag{3.30}$$

the ${}^I g_v = [0, 0, -g]^\top$ is the vector of gravity acceleration in the inertial frame (g is the value of local gravity acceleration).

Substituting the body velocity from (3.27) into T_1 , and after expanding the

vector inner product, there is

$$\begin{aligned}
T_1 &= \frac{1}{2} \sum_{i=0}^N m_i \dot{p}_i^\top \dot{p}_i \\
&= \frac{1}{2} \sum_{i=0}^N m_i (\dot{p}_C + J_{\omega i} \omega + J_{q i} \dot{q})^\top (\dot{p}_C + J_{\omega i} \omega + J_{q i} \dot{q}) \\
&= \frac{1}{2} M_t \dot{p}_C^\top \dot{p}_C + \frac{1}{2} \sum_{i=0}^N m_i p_{i/C}^\top p_C \\
&\quad + \frac{1}{2} \sum_{i=0}^N m_i (\omega^\top J_{\omega i}^\top J_{\omega i} \omega + 2\omega^\top J_{\omega i}^\top J_{q i} \dot{q} + \dot{q}^\top J_{q i}^\top J_{q i} \dot{q}).
\end{aligned} \tag{3.31}$$

Summation of $m_i p_{i/C}^\top p_C$ results in zero, because the identity in (3.12) shows

$$\frac{1}{2} \sum_{i=0}^N m_i p_{i/C}^\top p_C = \frac{1}{2} \left(\sum_{i=0}^N m_i p_{i/C} \right)^\top p_C = 0. \tag{3.32}$$

and thus,

$$T_1 = \frac{1}{2} M_t \dot{p}_C^\top \dot{p}_C + \frac{1}{2} \sum_{i=0}^N m_i (\omega^\top J_{\omega i}^\top J_{\omega i} \omega + 2\omega^\top J_{\omega i}^\top J_{q i} \dot{q} + \dot{q}^\top J_{q i}^\top J_{q i} \dot{q}). \tag{3.33}$$

The kinetic energy corresponding to rotation, which is the summation T_2 in (3.29), is expanded using (3.7), which yields

$$\begin{aligned}
T_2 &= \frac{1}{2} \sum_{i=0}^N ({}^i R \omega_i)^\top J_i ({}^i R \omega_i) \\
&= \frac{1}{2} \sum_{i=0}^N \omega^\top ({}^i R^\top J_i {}^i R) \omega + 2\omega^\top ({}^i R^\top J_i {}^i R F_i) \dot{q} + \dot{q}^\top (F_i^\top {}^i R^\top J_i {}^i R F_i) \dot{q}.
\end{aligned} \tag{3.34}$$

Combining (3.33) and (3.34) yields

$$\begin{aligned}
T = & \frac{1}{2} M_t \dot{p}_C^\top \dot{p}_C + \frac{1}{2} \omega^\top M_{\omega\omega} \omega \\
& + \omega^\top M_{\omega q} \dot{q} + \frac{1}{2} \dot{q}^\top M_{qq} \dot{q},
\end{aligned} \tag{3.35}$$

which is quadratic in variables \dot{p}_c , ω , and \dot{q} , where

$$\begin{aligned}
M_{\omega\omega} &= \sum_{i=0}^N m_i J_{\omega i}^\top J_{\omega i} + {}^i R_0^\top J_i^i R_0, \\
M_{\omega q} &= \sum_{i=0}^N m_i J_{\omega i}^\top J_{qi} + {}^i R_0^\top J_i^i R_0 F_i, \\
M_{qq} &= \sum_{i=0}^N m_i J_{qi}^\top J_{qi} + F_i^\top R_0^\top J_i^i R_0 F_i.
\end{aligned} \tag{3.36}$$

Defining the state vector

$$W = \begin{bmatrix} \dot{p}_C^\top & \omega^\top & \dot{q}^\top \end{bmatrix}^\top, \tag{3.37}$$

the kinetic energy can be concisely represented as

$$T = \frac{1}{2} W^\top M W, \tag{3.38}$$

where

$$M = \begin{bmatrix} M_t I_3 & O_{3 \times 3} & O_{3 \times N} \\ O_{3 \times 3} & M_{\omega\omega} & M_{\omega q} \\ O_{N \times 3} & M_{\omega q}^\top & M_{qq} \end{bmatrix} \tag{3.39}$$

can be seen as a generalized inertia matrix.

3.3.2 MOVA CM Translational Dynamics

The center of mass (CM) of the entire MOVA system is chosen as the representative point for describing the translational movement. The dynamics equation that governs p_C is derived by applying the Lagrangian method

$$\left(\frac{d}{dt} \frac{\partial L}{\partial \dot{p}_C} - \frac{\partial L}{\partial p_C} \right)^T = F_v, \quad (3.40)$$

where $F_v = [0, 0, F]^T$ is the thrust force vector acting on the VTOL aircraft body. Notice that transpose is applied on the left side of equation (3.40) to comply with numerator layout notation, which is required for the chain-rule to hold, *i.e.* $\dot{y} = \frac{\partial y}{\partial x} \dot{x}$, when x, y are vectors. The first term in the parenthesis on the left-hand side of (3.40) is expanded in (3.41), and the second term is evaluated in (3.42):

$$\begin{aligned} \left(\frac{d}{dt} \frac{\partial L}{\partial \dot{p}_C} \right)^T &= \frac{d}{dt} (M_t I_3 \dot{p}_C) \\ &= M_t \frac{d}{dt} (R_I^I \dot{p}_C) \\ &= M_t \left[R_I \frac{d}{dt} {}^I \dot{p}_C + \frac{d}{dt} (R_I)^I \dot{p}_C \right] \\ &= M_t [R_I \ddot{p}_C - S(\omega) R_I^I \dot{p}_C] \\ &= M_t I_3 \ddot{p}_C - M_t S(\omega) \dot{p}_C, \end{aligned} \quad (3.41)$$

$$\left(-\frac{\partial L}{\partial p_C} \right)^T = \left(\frac{\partial V}{\partial p_C} \right)^T = M_t \left(\frac{\partial (g_v^T R p_C)}{\partial p_C} \right)^T = M_t R_I g_v. \quad (3.42)$$

Combining (3.41) and (3.42) results in

$$M_t I_3 \ddot{p}_C - M_t S(\omega) \dot{p}_C + M_t R_I g_v = F_v. \quad (3.43)$$

3.3.3 Aircraft Rotational Dynamics

It is difficult to choose the configuration variables that capture 3D rotation of the aircraft body and facilitate the Lagrangian derivation. However, knowing that the Lagrangian L does not depend on the attitude of the aircraft body, it is feasible to derive the dynamics equation that governs the aircraft body rotation with a Lagrangian-like formulation that is equivalent to conservation of angular momentum, as

$$\left(\frac{d}{dt} \frac{\partial L}{\partial \omega} \right)^\top = \tau_a + \frac{\partial \dot{p}_0}{\partial \omega} F_v. \quad (3.44)$$

On the right side of the equation, the term τ_a is the external control torque applied to the body of the aircraft and the effect of the thrust force vector F_v to the ω -dynamics is $\frac{\partial \dot{p}_0}{\partial \omega} F_v = J_{\omega 0}^\top F_v$ using the principle of virtual work. Expanding $\frac{d}{dt} \frac{\partial L}{\partial \omega}$, there is

$$\begin{aligned} \left(\frac{d}{dt} \frac{\partial L}{\partial \omega} \right)^\top &= \frac{d}{dt} \left(\frac{\partial T}{\partial \omega} \right)^\top = \frac{d}{dt} (M_{\omega\omega}\omega + M_{\omega q}\dot{q}) \\ &= \frac{d^0}{dt} (M_{\omega\omega}\omega + M_{\omega q}\dot{q}) + S(\omega) (M_{\omega\omega}\omega + M_{\omega q}\dot{q}) \\ &= M_{\omega\omega} \frac{d}{dt} \omega + M_{\omega q} \ddot{q} + \frac{d}{dt} M_{\omega\omega} \omega + \frac{d}{dt} M_{\omega q} \dot{q} \\ &\quad + S(\omega) M_{\omega\omega} \omega + S(\omega) M_{\omega q} \dot{q}. \end{aligned} \quad (3.45)$$

Applying the identity $S(a)b = -S(b)a$, which results in

$$S(\omega) M_{\omega\omega} \omega = -S(M_{\omega\omega} \omega) \omega, \quad (3.46)$$

and the following rearrangement of the first term as

$$\frac{d}{dt} M_{\omega\omega} \omega = \left(\sum_{i=1}^N \dot{q}_i \frac{\partial M_{\omega\omega}}{\partial q_i} \right) \omega = \sum_{i=1}^N \dot{q}_i \frac{\partial M_{\omega\omega} \omega}{\partial q_i} = \frac{\partial M_{\omega\omega} \omega}{\partial q} \dot{q}, \quad (3.47)$$

there is,

$$\left(\frac{d}{dt}\frac{\partial L}{\partial \omega}\right)^\top = M_{\omega\omega}\frac{d}{dt}\omega + M_{\omega q}\ddot{q} + C_{\omega\omega}\omega + C_{\omega q}\dot{q}, \quad (3.48)$$

where

$$\begin{aligned} C_{\omega\omega} &= -S(M_{\omega\omega}\omega) + \frac{1}{2}\frac{d}{dt}M_{\omega\omega}, \\ C_{\omega q} &= \frac{1}{2}\frac{\partial M_{\omega\omega}\omega}{\partial q} + \frac{d}{dt}M_{\omega q} + S(\omega)M_{\omega q}. \end{aligned} \quad (3.49)$$

In fact, there are many other ways for factoring terms into $C_{\omega\omega}$ and $C_{\omega q}$ that will ensure the same the outcome of $C_{\omega\omega}\omega + C_{\omega q}\dot{q}$. The reason for this specific allocation is to preserve the skew-symmetric property of the final representation of the dynamics.

3.3.4 Joint Variables Dynamics

Joint variable dynamics can be evaluated with the same technique as the aircraft rotational dynamics using

$$\left(\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q}\right)^\top = \tau_m + \frac{\partial \dot{p}_0}{\partial \dot{q}}F_v, \quad (3.50)$$

where the vector $\tau_m = [\tau_{m1}, \tau_{m2}, \dots, \tau_{mN}]^\top$ represents the torque at each of the N revolute joints and $\frac{\partial \dot{p}_0}{\partial \dot{q}}F_v = J_{q0}^\top F_v$ represents the effect of F_v on the \dot{q} -dynamics

$$\begin{aligned} \left(\frac{d}{dt}\frac{\partial L}{\partial \dot{q}}\right)^\top &= \frac{d}{dt}(M_{\omega q}^\top \omega + M_{qq}\dot{q}) \\ &= \frac{d}{dt}(M_{\omega q}^\top)\omega + M_{\omega q}^\top \frac{d}{dt}\omega + M_{\omega q}^\top S(\omega)\omega + \frac{d}{dt}(M_{qq})\dot{q} + M_{qq}\ddot{q}, \end{aligned} \quad (3.51)$$

and

$$\left(-\frac{\partial L}{\partial q}\right)^\top = -\frac{1}{2} \left[\frac{\partial(M_{\omega\omega}\omega)}{\partial q}\right]^\top \omega - \left[\frac{\partial M_{\omega q}\dot{q}}{\partial q}\right]^\top \omega - \frac{1}{2} \left[\frac{\partial M_{qq}\dot{q}}{\partial q}\right]^\top \dot{q}. \quad (3.52)$$

Using similar method shown in (3.47) that transform between derivatives and partial derivatives, there is

$$\begin{aligned} \frac{d}{dt} M_{\omega q}^\top \omega &= \frac{\partial M_{\omega q}^\top \omega}{\partial q} \dot{q}, \\ \frac{d}{dt} M_{qq}^\top \dot{q} &= \frac{\partial M_{qq}^\top \dot{q}}{\partial q} \dot{q}, \end{aligned} \quad (3.53)$$

and

$$\left[\frac{\partial M_{\omega q}\dot{q}}{\partial q}\right]^\top \omega = \left[\frac{\partial M_{\omega q}^\top \omega}{\partial q}\right]^\top \dot{q}. \quad (3.54)$$

Invoking these identities, (3.51) and (3.52) sum to form the left-hand side of (3.50)

as

$$M_{qq} \frac{d}{dt} \omega + M_{qq} \ddot{q} + C_{q\omega} \omega + C_{qq} \dot{q} = \tau_m + J_q F_v, \quad (3.55)$$

where

$$\begin{aligned} C_{q\omega} &= (-S(\omega)M_{\omega q})^\top - \frac{1}{2} \left[\frac{\partial(M_{\omega\omega}^\top \omega)}{\partial q}\right]^\top, \\ C_{qq} &= \frac{\partial M_{\omega q}^\top \omega}{\partial q} - \left[\frac{\partial M_{\omega q}^\top \omega}{\partial q}\right]^\top + \frac{\partial M_{qq}^\top \dot{q}}{\partial q} \dot{q} - \frac{1}{2} \left[\frac{\partial M_{qq}^\top \dot{q}}{\partial q}\right]^\top \dot{q}. \end{aligned} \quad (3.56)$$

Derivatives and partial derivatives of $M_{\omega\omega}$, $M_{\omega q}$, and M_{qq} can be obtained using the chain-rule on their definition in (3.36), which results in expression of derivatives and partial derivatives of $J_{\omega i}$, J_{qi} , F_i and R_i as follows:

$$\frac{d}{dt} J_{\omega i} = S(J_{qi} \dot{q}) = \sum_{j=1}^N S(J_{qi} \delta j) \dot{q}_j \quad (3.57)$$

$$\frac{\partial}{\partial q_j} = S(J_{qi}\delta_j) \quad (3.58)$$

$$\frac{d}{dt}F_i = [O_{3 \times 1}, -S(R_1 u_2)F_1 \dot{q}, \dots, -S(R_{i-1} u_i)F_{i-1} \dot{q}, O_{3 \times (N-i)}] \quad (3.59)$$

$$\frac{\partial}{\partial q_j} F_i = \begin{cases} [O_{3 \times j}, -S(R_j u_{j+1})(R_{j-1} u_j), \dots, -S(R_{i-1} u_i)(R_{j-1} u_j), O_{3 \times (N-i)}], & j < i; \\ O_{3 \times N}, & j \geq i. \end{cases} \quad (3.60)$$

$$\frac{d}{dt}J_{qi} = - \sum_{j=1}^N S(S(F_j \dot{q})R_j^j v_{ij}) F_j + S(R_j^j v_{ij}) \frac{d}{dt}F_j \quad (3.61)$$

$$\frac{\partial}{\partial q_k} = - \sum_{j=1}^N S(S(F_j \delta_k)R_j^j v_{ij}) F_j + S(R_j^j v_{ij}) \frac{\partial}{\partial q_k} F_j \quad (3.62)$$

$$\frac{d}{dt} {}^i R_0^\top J_i^i R_0 = \frac{d}{dt} R_i J_i R_i^\top = S(F_i \dot{q}) R_i J_i R_i^\top - R_i J_i R_i^\top S(F_i \dot{q}) \quad (3.63)$$

$$\frac{\partial}{\partial q_j} {}^i R_0^\top J_i^i R_0 = \begin{cases} S(R_{j-1} u_j) R_i J_i R_i^\top - R_i J_i R_i^\top S(R_{j-1} u_j), & j \leq i; \\ O_{3 \times 3}, & j > i. \end{cases} \quad (3.64)$$

Column selector δ_i is defined as a column vector of appropriate dimension with only the i th entry being 1 and others being 0, so that $A\delta_i$ will result in the i th column of matrix A .

3.3.5 MOVA Dynamics in General Robotic Form

It is now possible to assemble the dynamics equation from the three parts (in Sections 3.3.2, 3.3.3, and 3.3.4) into the standard form dynamics equation widely used in control and robotics research

$$M\dot{W} + CW + G = \tau, \quad (3.65)$$

where ω is the MOVA state vector defined in (3.37), M is defined in (3.39) and

$$\begin{aligned} C &= \begin{bmatrix} M_t S(\omega) & O_{3 \times 3} & O_{3 \times N} \\ O_{3 \times 3} & C_{\omega\omega} & C_{\omega q} \\ O_{N \times 3} & C_{q\omega} & C_{qq} \end{bmatrix}, \\ G &= \begin{bmatrix} (M_t R_I g_v) \\ O_{3+N} \end{bmatrix}, \\ \tau &= \begin{bmatrix} F_v \\ \tau_a + J_{\omega 0}^\top F_v \\ \tau_m + J_{q 0}^\top F_v \end{bmatrix} = BU, \end{aligned} \quad (3.66)$$

where

$$B = \begin{bmatrix} I_3 & O_{3 \times 3} & O_{3 \times 3} \\ J_{\omega 0}^\top & I_3 & O_{3 \times 3} \\ J_{q 0}^\top & O_{N \times 3} & I_N \end{bmatrix} \text{ and } U = \begin{bmatrix} F_v \\ \tau_a \\ \tau_m \end{bmatrix}. \quad (3.67)$$

3.3.6 Skew-symmetric Property of the Dynamics Equation

Skew-symmetric property of the dynamics equation is related to the passivity of the system and conservation of energy, and is thus referred as “passivity” property,

which is an important property for controller development [41]. Let

$$\begin{aligned}\Delta &= \frac{d}{dt}M - 2C \\ &= \begin{bmatrix} \Delta_p & O_{3 \times 3} & O_{3 \times N} \\ O_{3 \times 3} & \Delta_{\omega\omega} & \Delta_{\omega q} \\ O_{N \times 3} & \Delta_{q\omega} & \Delta_{qq} \end{bmatrix},\end{aligned}\tag{3.68}$$

the skew-symmetric property demands

$$\Delta - \Delta^\top = 0.\tag{3.69}$$

Calculating the submatrices one by one, there is

$$\begin{aligned}\Delta_p &= -M_t S(\omega), \\ \Delta_{\omega\omega} &= \frac{1}{2} \frac{d}{dt} M_{\omega\omega} + S(M_{\omega\omega} \omega) - \frac{1}{2} \frac{d}{dt} M_{\omega\omega} = S(M_{\omega\omega} \omega), \\ \Delta_{\omega q} &= \frac{1}{2} \frac{d}{dt} M_{\omega q} - \frac{1}{2} \frac{\partial M_{\omega\omega} \omega}{\partial q} - \frac{d}{dt} M_{\omega q} - S(\omega) M_{\omega q} \\ &= -\frac{1}{2} \frac{d}{dt} M_{\omega q} - \frac{1}{2} \frac{\partial M_{\omega\omega} \omega}{\partial q} - \frac{d}{dt} M_{\omega q} - S(\omega) M_{\omega q}, \\ \Delta_{q\omega} &= \frac{1}{2} \frac{d}{dt} M_{\omega q}^\top + (S(\omega) M_{\omega q})^\top + \frac{1}{2} \left[\frac{\partial (M_{\omega\omega}^\top \omega)}{\partial q} \right]^\top, \\ \Delta_{qq} &= \frac{1}{2} \frac{d}{dt} M_{qq}^\top - \frac{\partial M_{\omega q}^\top \omega}{\partial q} + \left[\frac{\partial M_{\omega q}^\top \omega}{\partial q} \right]^\top - \frac{\partial M_{qq} \dot{q}}{\partial q} + \frac{1}{2} \left[\frac{\partial M_{qq} \dot{q}}{\partial q} \right]^\top \\ &= -\left(\frac{\partial M_{\omega q}^\top \omega}{\partial q} - \left[\frac{\partial M_{\omega q}^\top \omega}{\partial q} \right]^\top \right) - \frac{1}{2} \left(\frac{\partial M_{qq} \dot{q}}{\partial q} - \left[\frac{\partial M_{qq} \dot{q}}{\partial q} \right]^\top \right).\end{aligned}\tag{3.70}$$

As Δ_p , $\Delta_{\omega\omega}$ and Δ_{qq} are skew-symmetric, and $\Delta_{\omega q} = -\Delta_{q\omega}^\top$, the overall matrix Δ satisfies (3.69).

3.4 Automation of Dynamics Derivation

Dynamics equation for general MOVA system is compacted into a concise and widely known form in (3.65), which is suitable for analysis and other derivative work. However, evaluation of explicit terms of the model is still necessary when implementing a MOVA simulator or model-based controller, it is necessary to follow the steps in Sec. 3.3 and find the explicit expression of the dynamics model. One obstacle in doing this is that number of terms in the explicit dynamics equation grows super-linearly with respect to the number of rigid bodies in the system. The size of the MOVA dynamics model expression grows quickly as the number of links, N , increases. Even when N is small, for example 2, the complexity of the explicit terms in the dynamics equation causes manual derivation to be tedious and error-prone.

MOVADYN, a computer program that derives the MOVA dynamics equation automatically, is developed as a solution for this issue. Auxiliary tools are also created to interface MOVADYN with up- and down-stream of design steps, enabling automated flow of design information from 3D modeling of the system to executable simulation or controller code. In this section, MOVADYN and auxiliary tools are briefly introduced and demonstration of usage of these tools are shown with examples at the end of this section.

3.4.1 MOVADYN Dynamics Derivation Program

MOVADYN is written for general MOVA systems with a serial onboard manipulator with any number of links connected by revolute joints, as initially as stated in Sec. 3.2. Given sufficient input to describe the kinematics structure and the mass distribution of the system, this program is able to perform calculation without human intervention and find explicit expressions for matrices and vectors M , C , G and τ in

(3.65). Designed to run in a symbolic calculation software package Mathematica[®], it is able to process parametric models (physical parameters input in symbolic form). Users of MOVADYN have the freedom to choose whether to substitute in exact values of the physical parameters before or after the dynamics derivation. Using numerical values will end up with a faster derivation process, but results in less information revealed in the output, since all the numbers are combined together and it is almost impossible to see which physical parameters are most influential. On the other hand, if the symbolic form of parameters are input to MOVADYN, the output dynamics equation contains the same set of symbols as the input, which enables evaluation of the impact of certain parameters on the resulting system dynamics equation.

The following set of input parameters are required to run the MOVADYN program (for $1 \leq i \leq N$): mass, m_i , and moment of inertia at principal axes, J_i , of each link; dimension information of each link in terms of ${}^i r_i$ and ${}^i l_i$ vectors; rotation axis u_i of each joint, and pose of each link relative to the aircraft body when the manipulator configuration vector $q = 0$, which is $R_{i0} = R_i|_{q=0}$, assuming the body frames of the VTOL aircraft and each link are aligned with the principle axis of the associated rigid body./ Output involves kinematics calculation are also included in the result for convenience in addition to matrices and vectors that constitute the dynamics equation (3.65).

Implementation of the MOVADYN program follows the steps of derivation described in Section 3.2 and 3.3. The entire derivation process is shown in a flow chart in Fig.3.2. Kinematics related matrices and vectors are first evaluated as preparation for the dynamics model. Vectors ${}^j v_{ij}$ are evaluated by (3.16) using the mass of each individual links, m_i , and ${}^i l_i$ and ${}^i r_i$ as in input. Rotation matrices, ${}^{i-1} R_i$, representing transformation between frames attached to adjacent bodies, are calculated using the rotation axis of the revolute joints u_i . Rotation matrices R_i are found by matrix

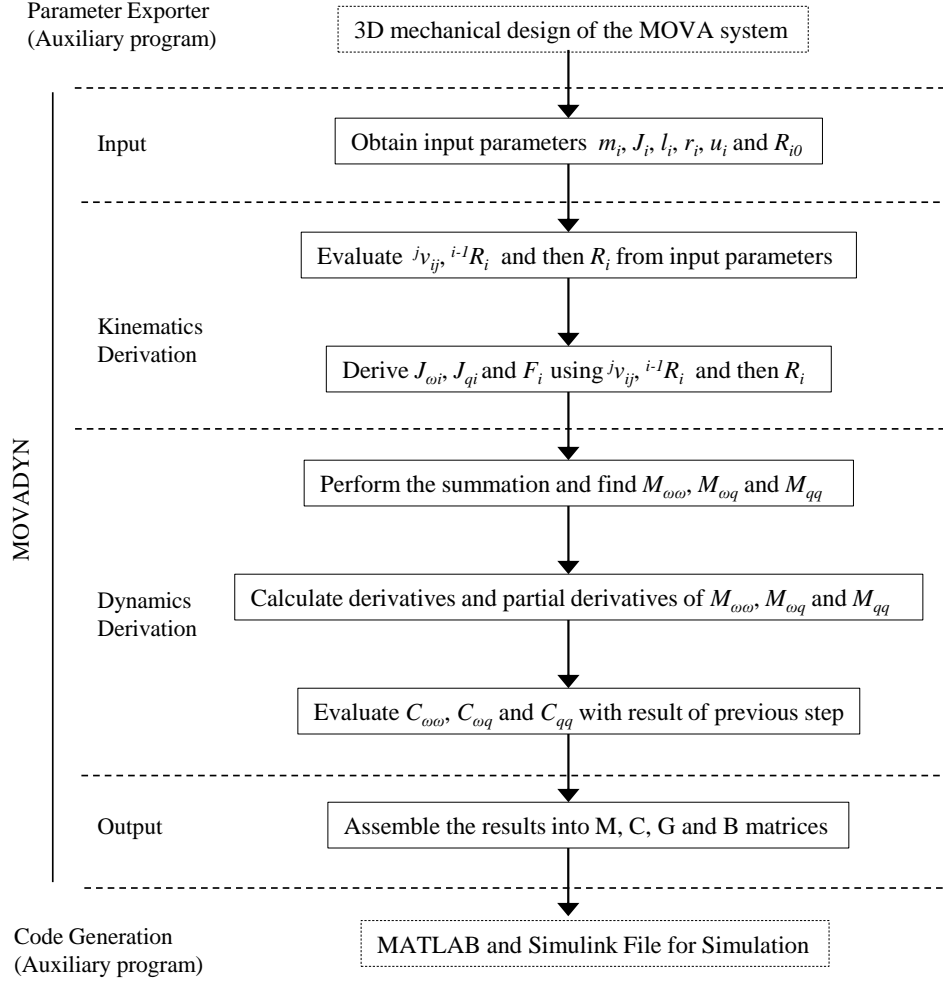


Figure 3.2: Flow chart of MOVADYN and its auxiliary program

multiplication of ${}^{i-1}R_i$ by (3.3). Then, F_i , the matrix that relates joint velocity to angular velocity of each body relative to the aircraft body, is derived by (3.8). Jacobian matrices for body i , $J_{\omega i}$ and J_{qi} , are calculated using (3.28).

Based on matrices $J_{\omega i}$ and J_{qi} found in the kinematics above, parts of the general inertial matrices, $M_{\omega\omega}$, $M_{\omega q}$ and M_{qq} , are calculated by (3.36) and later assembled into M . Time derivatives of $M_{\omega\omega}$, $M_{\omega q}$ and M_{qq} and their partial derivatives with respect to q are found using appropriate symbolic calculus functions in Mathematica, the symbolic software. Matrices $C_{\omega\omega}$, $C_{\omega q}$, $C_{q\omega}$ and C_{qq} are calculated by (3.49) and (3.56), which then leads to construction of C matrix by (3.66). Gravity matrix G and input vector τ are easy to find following their expression in (3.66).

Kinematics related matrices $J_{\omega i}$ and J_{qi} are often necessary for complete simulation of the MOVA system and are thus included in the derivation result in addition to matrices and vectors M , C , G and τ in the dynamics equation.

3.4.2 Auxiliary Interfacing Tools

MOVADYN will output human readable dynamics equation expressions on the screen with input of system physical parameters. However, a MOVA system design, like other mechanical systems, starts with structural design and finally requires executable code for implementing dynamics simulation and controller testing. In order to fit the MOVADYN dynamics derivation program into a practical MOVA system development scenario, two auxiliary tools are made: i) automatically transfer information from the mechanical design software to physical parameter input needed by the MOVADYN program; and ii) output dynamics matrix into ready-to-run code for dynamics simulation and controller implementation. These tools connect the upstream and downstream of the dynamics derivation and greatly streamline the

development process by allowing faster and tighter iteration in the design process.

3.4.2.1 MOVA Code Generation Utility

The MOVADYN program derives the dynamics equation of a MOVA system. The resulting mathematical expressions are expected to be complex so that transforming them into executable code that can be included in simulation or controller programs is a tiresome job. For this reason, an automatic code generation utility is included in MOVADYN. The code generated is output in the form of MATLAB functions and Simulink models which are widely used in both academics and industry and can be later translated into other languages or directly into executable binary using tools provided by Mathworks. The code generation work flow is illustrated in Fig. 3.3. The code generation utility converts Mathematica expressions into code in text form that is recognized by the MATLAB interpreter and then inserts the code inside customized template files so that the resulting files are complete MATLAB functions or Simulink model files. These files include MATLAB functions that evaluate dynamics and kinematics of the corresponding MOVA system and a script that loads and saves system parameters. In addition, a Simulink model that contains a monolithic block that represents the MOVA system is also generated. This block invokes the dynamics and kinematics for simulation of the MOVA system plant and provides a convenient encapsulation with input and output ports defined so that it can be integrated into a higher level system.

The code generation utility enables any change in design of the MOVA system to be propagated, with minimal manual intervention, to executable code which can be used for simulation and controller implementation. It basically automates the downstream use of the MOVADYN program.

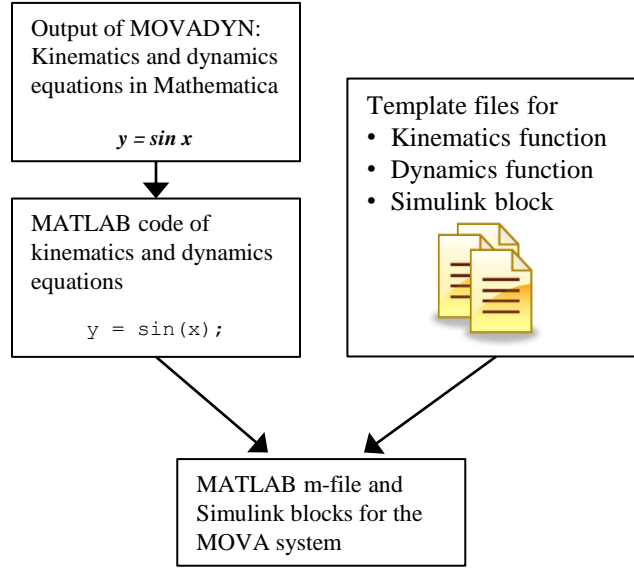


Figure 3.3: Illustration of MOVADYN code generation facility

3.4.2.2 Physical Parameter Exporter

The parameter exporter is another auxiliary program of MOVADYN. It enables the upstream automation of MOVADYN by extracting physical parameters from a MOVA system mechanical design and puts them into a script that calls MOVADYN to perform derivation of dynamics and later generation of executable code. This utility supports Autodesk Inventor, one of the leading 3D mechanical design tools, at present, and is expected to support other software in the future. Inertial and dimensional parameters of the designed mechanical system as well as information about kinematics constraint that represents joints between links can be extracted with this exporter. The outcome is summarized into a script file acceptable by Mathematica, which is also the symbolic calculation software that MOVADYN and code generator requires. This way, a top level script can call the generated parameter script file, MOVADYN, and the code generator consecutively to propagate design information updates from the source, which is in the 3D mechanical model, into code ready to be

integrated into a dynamics simulator or a model-based controller.

3.4.3 Usage and Example

The software is provided as open source and anyone can download these tools from [42] to use with a MOVA project or other relevant projects. It is licensed under a Creative Commons Attribution 4.0 International License [43]. Modifications and improvement of these tools are welcome as long as the original author and copyright information are retained.

MOVADYN and the accompanying code generation utility are packed in a single Mathematica package, called “MOVADynamics”. The code generation utility requires a third-party Mathematica package named “ToMatlab” by Harri Ojanen, which is available from Wolfram Library Archive [44]. Instructions for installation of the MOVADYN package in Mathematica can be accessed via the online help file. After the package is installed, the following code imports the package so its content can be used:

```
1 << MOVADynamics';
```

MOVADYN, which is the core dynamics derivation functionality, can be invoked by the following line,

```
1 Dyn = DeriveMovaDynamics[Nlink, Ms, Is, ls, rs, Rs, vs];
```

where variables Nlink, Ms, Is, ls, rs, Rs and vs are N , m_i , J_i , ${}^i l_i$, ${}^i r_i$, ${}^{i-1} R_i|_{q=0}$, and u_i , respectively. Per the conventions of the symbolic algebra software being used, parameters of the same nature but for different links are grouped into an ordered list using curly brackets. These parameters are passed into the derivation program **DeriveMovaDynamics**. Results of the dynamics derivation are retrieved from the

Table 3.2: Names of matrices and vectors in MOVADYN output

Name	Corresponding symbol
Nlink	N
M	M
C	C
T	R_i
Gb	g_v
B	B
FwdKin	p_i

returned value Dyn. For example, Dyn[“C”] contains the C matrix in MOVA dynamics equation. A list of the variable name and the corresponding matrices or vectors is shown in Table 3.2.

The MOVA code generation utility substitutes placeholders in the template files with the code embodies the kinematics and dynamics equations. Thus, in order for code generation utility to function properly, a set of custom template files are to be provided. Both the kinematics and dynamics template files have three parts, the preamble, placeholder, and epilogue. The preamble contains the function definition, description of the function and extra statements that convert the form of the input parameters so that it can be used in the calculations. The placeholder in the dynamics template file is a line with “%//[AUTO_GEN_DYNAMICS]”. It will be replaced by the code that represents the corresponding equations, in which the following variables are referenced: “qd1”, “qd2”, ... “qdN” which represents \dot{q}_i ($1 < i < N$); “q”, a vector that corresponds to q ; and “w0x”, “w0y”, “w0z” which are elements of the body angular velocity “ ω ”. The code that replaces the place holder uses these variables and the physical parameters of the system to calculate M , C , G and B matrices. The place holder section of kinematics template files are laid out similarly. The differences are that the placeholder is a single line “%//[AUTO_GEN_KINEMATICS]” and for the a single kinematics template functions, two MATLAB functions are generated,

one yields the rotation matrix of each body in the inertial frame and the other outputs the center of mass of each body relative to the inertial origin. In the epilog section, additional code can be placed for post-processing purposes, e.g. \dot{W} can be calculated with the matrices and input vector in the dynamics template function.

MATLAB functions that relate to accessing the physical parameters are generated at the same time. There are three of these functions, a parameter pack function, a parameter unpack function and a parameter definition function. The pack and unpack functions cooperate with the kinematics and dynamics functions to fulfill the calculation and do not need further editing after being generated. The parameter definition function will define the actual numerical values for all the symbolic parameters used during MOVADYN derivation. The generated definition function will contain every parameter that needs to be specified but assign them with zeros. The user have to change them to the actual value in SI units, e.g. the mass is defined with unit of kilogram and length in meter.

In addition, Simulink blocks of the MOVA system and a few helper m-files for accessing the parameters are also be generated by the code generator. The generated Simulink block conveniently added into a higher level Simulink diagram with input signal and logging facility to complete a simulation test bench. The following code snippet demonstrate the syntax used for code generation: “Id” is a string that contains a unique name. This name will be postfix to all generated MATLAB function names as to avoid name collision; “TmpDir” is the directory that contains template files; “OutDir” is the directory where the generated files are written be and “Dyn” is the output of the MOVADYN dynamics derivation program.

```

1 (* Write out Dynamics and Kinematics MATLAB functions *)
2 WriteMovaDynamics[Id, TmplDir, OutDir, Dyn];
3 (* Write out Parameter MATLAB functions *)
```

```

4 WriteMovaParameterFunction[Id, OutDir, Dyn];
5 (* Write out Simulink block *)
6 WriteMovaSimulinkBlock[Id, TmplDir, OutDir];

```

An complete example of using MOVADYN program to derive the dynamics equation of the planar MOVA system described in Chapter 2 is given in Appendix A.1.

The physical parameter exporter is provided as a macro (named “MOVA-DynamicsExport”) which can be loaded in Autodesk Inventor. In order to use the exporter, a 3D model of the designed MOVA system have to be built. The 3D model should be present as a part assembly with aircraft body and each link being modeled as a “part”. Revolute joints that connects links and the aircraft body in serial have to be modelled with the “Insert” constraint. Additional constraints can be added in order to place the links into a zero q configuration (the relative pose when $q = 0$). To generate the Mathematica script that contains extracted physical parameters, execute the macro with the “part” that represent aircraft body being selected (highlighted). A file will be generated in the same directory as the 3D model assembly file with “_mdef” as postfix in file name. An example of using parameter exporter is in Appendix A.1.

3.5 Validation of Derived Dynamics Equation

In this section, results from the MOVADYN automatic dynamics equation derivation tool are verified Besides rigorously checking derivation steps in Sec. 3.2 and Sec. 3.3 and the fidelity of the automatic derivation program implementation, two distinct forms of cross-validation procedures are performed to ensure the correctness of the MOVADYN output. For a planar MOVA system, the symbolic form

of derivation output is directly compared with manually derived dynamics equation to show equivalence between the two. For a more sophisticated MOVA system in 3D space, validation is conducted by comparing the responses of two simulation of the same MOVA system. One simulation is constructed using dynamics derivation output of MOVADYN, and the other simulation is based on a off-the-shelf numerical dynamics simulator.

3.5.1 Cross-validation with Manual Derived Dynamics Equations

First, the resulting dynamics model from MOVADYN is compared with manually derived dynamics equation of the same system. MOVA systems with multi-link onboard manipulator in 3D space will result in intricate dynamics equations which renders manually derivation impractical. For such system, it is very time consuming to derive dynamics by hand and conceding that it is accomplished the correctness of the steps in manual derivation will be equivalently questionable due to the complexity. Thus, the planar MOVA system described in Chap. 2 is chosen. Hand-derived dynamics of the planar MOVA system is shown in equations (2.21), (2.22), and (2.25) in Sec. 2.3. The script that derives dynamics for the planar MOVA using MOVADYN program is listed in Sec. 3.4.3.

The derivation result of MOVADYN program needs further processing before comparing to (2.21), (2.22), and (2.25) because they are based on slightly different setup. MOVADYN program is designed for MOVA system in 3D space and results in dynamics equation written in body frame, however, the planar MOVA system is constrained in 2D space and does not have dynamics for translation and rotation in the third dimension and the manually derived dynamics equation are described in

the inertial frame. In order for a valid comparison, the result from MOVADYN is transformed into the inertial frame and variables are replaced to fit those used in Chap. 2.

First, projecting the output of MOVADYN into a planar world in the x-z plane, components corresponding to y-axis of \dot{p}_c , x- and z-axis of ω do not exist. The state variable becomes

$$W' = \begin{bmatrix} \dot{p}_{Cx} & \dot{p}_{Cz} & \omega_y & \dot{q}_1 \end{bmatrix}^T \quad (3.71)$$

with the corresponding dynamics equation being

$$M' \frac{d}{dt} W' + C' W' + G' = \tau'. \quad (3.72)$$

Rows and columns 2,4 and 6 in M , C , G , τ are deleted and result in M' , C' , G' , τ' , which are explicitly written as

$$M' = \begin{bmatrix} m_0 + m_1 & 0 & 0 & 0 \\ 0 & m_0 + m_1 & 0 & 0 \\ 0 & 0 & J_0 + J_1 + \frac{l_1^2 m_0 m_1}{4(m_0 + m_1)} & J_1 + \frac{l_1^2 m_0 m_1}{4(m_0 + m_1)} \\ 0 & 0 & J_1 + \frac{l_1^2 m_0 m_1}{4(m_0 + m_1)} & J_1 + \frac{l_1^2 m_0 m_1}{4(m_0 + m_1)} \end{bmatrix} \quad (3.73)$$

$$C' = \begin{bmatrix} 0 & (m_0 + m_1)\omega_y & 0 & 0 \\ -(m_0 + m_1)\omega_y & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.74)$$

$$G' = R^T \begin{bmatrix} 0 & (m_0 + m_1)g & 0 & 0 \end{bmatrix}^T \quad (3.75)$$

$$\tau' = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ \frac{l_1 m_1 \sin q_1}{2(m_0 + m_1)} & 1 & 0 \\ \frac{l_1 m_1 \sin q_1}{2(m_0 + m_1)} & 0 & 1 \end{bmatrix} \begin{bmatrix} F \\ \tau'_a \\ \tau'_m \end{bmatrix}. \quad (3.76)$$

The first two rows of (3.72) are extracted and form the translational equation

$$m_T \frac{d}{dt} \dot{p}_{C2} + m_T S(\omega_y) \dot{p}_{C2} + m_T R^\top g_v = F_v \quad (3.77)$$

where $\dot{p}_{C2} = [\dot{p}_{Cx}, \dot{p}_{Cz}]^\top$, $S(\omega_y) \in \mathfrak{so}(2)$ is skew-symmetric matrix in 2D space, $R \in SO(2)$ represents a transformation from the aircraft body frame to the inertial frame, $F_v = [0, F]^\top$, and $g_v = [0, g]^\top$. Since \dot{p}_{C2} is a vector in the aircraft body frame, there is

$$\begin{aligned} \frac{d}{dt} \dot{p}_{C2} &= \frac{d}{dt} (R^{\top I} \dot{p}_{C2}) \\ &= -S(\omega_y) R^{\top I} \dot{p}_{C2} + R^{\top I} \ddot{p}_{C2} \\ &= -S(\omega_y) \dot{p}_{C2} + R^{\top I} \ddot{p}_{C2}. \end{aligned} \quad (3.78)$$

Substituting (3.78) into (3.77), and left-multiplying both sides by R , yields

$$m_T^I \ddot{p}_{C2} + m_T g_v = R F_v, \quad (3.79)$$

which is equivalent to (2.21). Subtracting the fourth row of (3.72) from the third row, yields

$$J_0 \dot{\omega}_y = \tau'_a - \tau'_m. \quad (3.80)$$

Taking the fourth row out of (3.72), results in

$$\left(J_1 + \frac{l_1^2 m_0 m_1}{4m_T}\right)(\omega_y + \dot{q}_1) = \frac{l_1 m_1 \sin q_1}{2m_T} F + \tau'_m. \quad (3.81)$$

Comparing the assignment of frames and definitions of variables in MOVADYN and the manual derivation in Chapter 2, there are

$$\begin{aligned} {}^I\ddot{p}_{C2} &= \ddot{p}_{vg} \\ \tau'_a &= -\tau_0, \\ \tau'_m &= -\tau_1, \\ \omega_y &= -\dot{\theta}_0, \\ q_1 &= -\theta_1 + \pi/2. \end{aligned} \quad (3.82)$$

Thus, (3.79) is equivalent to

$$m_T \ddot{p}_{vg} + m_T g_v = \begin{bmatrix} -\sin \theta_1 \\ \cos \theta_1 \end{bmatrix} F, \quad (3.83)$$

(3.80) can be rewritten as

$$J_0 \ddot{\theta}_0 = \tau_0 - \tau_1, \quad (3.84)$$

and (3.81) can be transformed into

$$\bar{J}_1 \ddot{\theta}_{01} = \tau_1 - \frac{l_1 m_1 \cos \theta_1}{2m_T} F, \quad (3.85)$$

using the trigonometry identity $\sin q_1 = \sin(-\theta_1 + \pi/2) = \cos \theta_1$. It is concluded that the dynamics equation output from MOVADYN is identical to that from manual algebraic derivation because (3.83), (3.84) and (3.85) are identical to (2.21), (2.22)

and (2.25), respectively.

3.5.2 Cross-validation with Numerical Dynamics Simulator

Another validation is performed via comparing the simulation result of the derived dynamics equation and that of the same system with a numerical dynamics simulator under identical excitation. Numerical multi-body dynamics simulation software is used to provide ground truth. These software packages take kinematics constraints and physical properties of the system as input and generate a numerical representation of the dynamics system behavior under a certain initial condition and input in the form of forces and torques.

The elements of the validation test are illustrated in Fig. 3.4. The test input signal is fed into two different simulations, first of which is based on a numerical dynamics simulation (I); the second is based on dynamics equation generated from MOVADYN (II). By comparing the output of both simulations, fidelity of the derived model can be evaluated. SimMechanics[®] toolbox of Simulink[®] was used as the numerical dynamics simulator [21]. Position and pose of the end-effector in the inertial frame is chosen as the output of the simulation. A rubric of comparison includes two scalars: position error, which is the norm of end-effector position vector difference, and the pose error, which is defined as the minimum rotation angle that is able to abridge the discrepancy between pose of the end-effector in the two simulations. Small values for both scalars after a sufficient period of simulation time indicates the closeness of the simulation outputs, which signifies the correctness of the MOVADYN derivation result. Note that although it is impossible to cover every input case and finally show the correctness of the derivation result, conclusive results from a few distinct input sequences will greatly improve the confidence in the MOVADYN output.

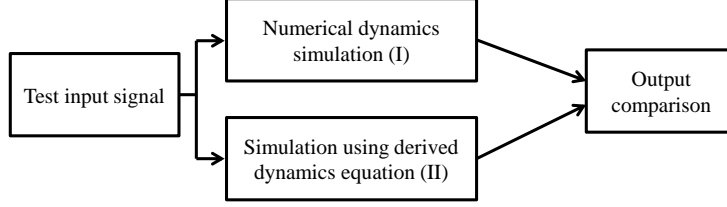


Figure 3.4: Block diagram of dynamics validation test-bed.

The MOVA system used for validation has a 2DOF onboard manipulator (see Fig. 3.6). The VTOL aircraft has mass m_0 and moment of inertia J_0 . The first and the second joints are co-located at the center of mass of the VTOL, leaving the first link of the manipulator massless. Rotation axis of Joint 1 is aligned with the y-axis of the VTOL aircraft and the rotation axis of Joint 2 is in line with x-axis of frame \mathcal{B}_1 . The second link of the manipulator is modeled as a long and thin cylinder with uniform density with mass and moment of inertia as m_a and J_0 , respectively. The length of this link is noted as l_a . Pose of the manipulator link is pointing upward if manipulator configuration vector $q = [0, 0]^T$ [rad]. Physical parameters values of this MOVA system are listed in Table 3.3. Note that the mass and moment of inertia of the VTOL aircraft body and the onboard manipulator are comparable.

The entire simulation is scaffolded in Simulink. The SimMechanics simulation is one sub-system of the top level simulation diagram, and is shown in Fig. 3.5. The sub-system representing the simulation using MOVADYN is directly generated from the MOVADYN and its accessory program.

Among many input signal test cases simulated, results from three representative test case are presented. The initial condition is set as $p_c(0) = [0, 0]^T$ [m], $R_0(0) = I_3$ and $q(0) = [\pi, 0]^T$ [rad], *i.e.* the CM of the entire system is placed at the origin, and VTOL aircraft remains level with the manipulator link pointing down,

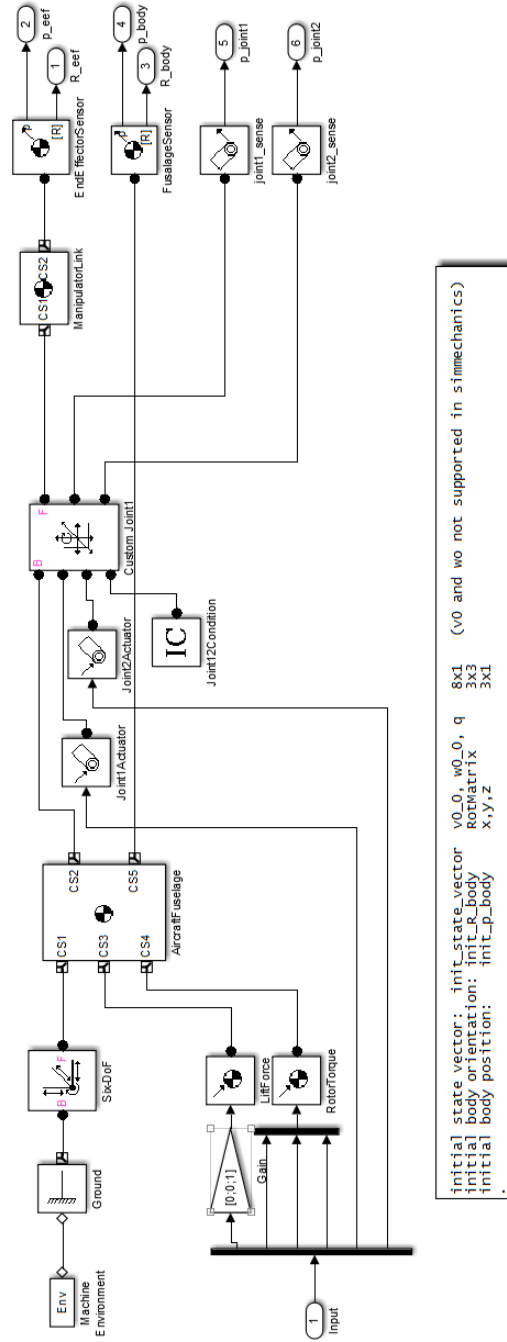


Figure 3.5: Simulink diagram of the SimMechanics simulation.

Table 3.3: Physical Parameters of the 3D MOVA system used for validation.

Parameter	Value
m_0	2 kg
J_0	$\text{diag}(\frac{1}{6}, \frac{1}{6}, \frac{1}{3}) \text{ kg} \cdot \text{m}^2$
m_a	1 kg
J_a	$\text{diag}(\frac{1}{48}, \frac{1}{48}, \frac{1}{800}) \text{ kg} \cdot \text{m}^2$
l_a	0.5 m

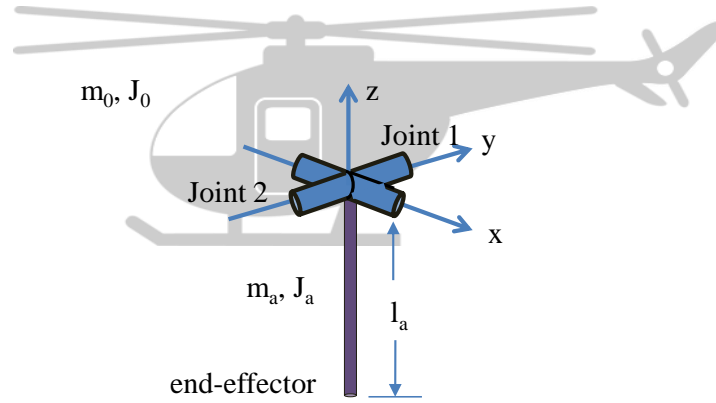


Figure 3.6: Illustration of a 3D MOVA system with 2DOF onboard manipulator.

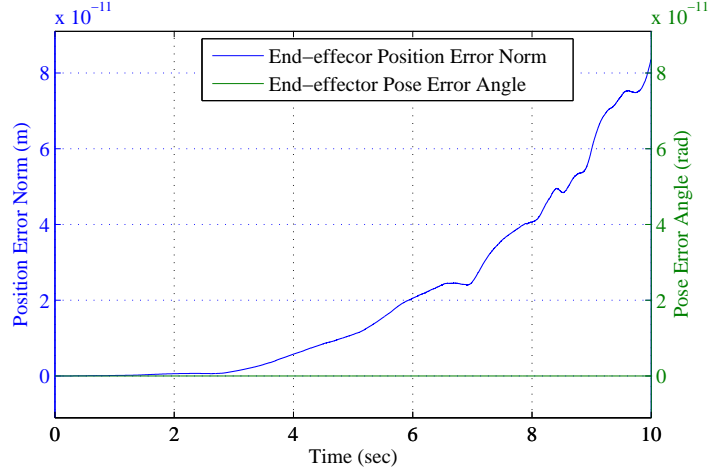


Figure 3.7: Plot of dynamics validation result with random input signals.

except for third case $q(0) = [\frac{\pi}{2}, 0]^T$ [rad].

In the first test case, random noise signals were used as the input test signals. All three input signals, $F(t)$, $\tau_a(t)$, and $\tau_m(t)$, are piece-wise constant with values determined by the corresponding discrete time signals, $F[n]$, $\tau_a[n]$, and $\tau_m[n]$, with sample time $T_s = 0.1$ sec,

$$\begin{bmatrix} F(t) \\ \tau_a(t) \\ \tau_m(t) \end{bmatrix} = \begin{bmatrix} F[n] \\ \tau_a[n] \\ \tau_m[n] \end{bmatrix} \quad (nT_s \leq t < (n+1)T_s). \quad (3.86)$$

Discrete time signals $F[n]$ and elements of $\tau_a[n]$ and $\tau_m[n]$ follow uniform distributions with zero means, $F[n] \sim U(0, 50)$ N, $\tau_{ai}[n] \sim U(-1, 1)$ and $\tau_{mj}[n] \sim U(-0.2, 0.2)$, where τ_{ai} and τ_{mj} are elements of τ_a and τ_m , respectively, $1 \leq i \leq 3$ and $1 \leq j \leq 2$. The simulation was run for 10 secs and the result is shown in Fig. 3.7. The position error is on the order of 1×10^{-9} [m] and the pose error is smaller than 1×10^{-8} [rad], which is the smallest angle that can be identified by the minimum rotation angle algorithm.

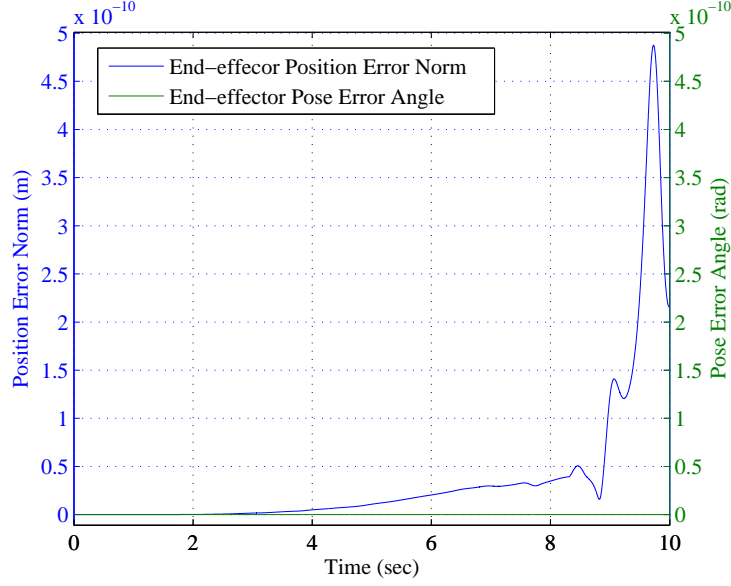


Figure 3.8: Plot of dynamics validation result with the sine wave test input signal.

In the second case, sinusoidal input signals are fed into the simulations. The input signals, $F(t)$, $\tau_a(t)$, and $\tau_m(t)$, are generated according to

$$\begin{aligned}
 F(t) &= m_T g + 5 \sin \pi t \text{ N} \\
 \tau_a(t) &= \begin{bmatrix} 0.1 \sin \pi t \\ 0.5 \sin(2\pi t + \frac{1}{2}\pi) \\ -0.5 \sin(\pi t + \frac{1}{3}\pi) \end{bmatrix} \text{ N} \cdot \text{m} \\
 \tau_m(t) &= \begin{bmatrix} 0.5 \sin(2\pi t + \frac{1}{2}\pi) \\ 0.1 \sin \pi t \end{bmatrix} \text{ N} \cdot \text{m}
 \end{aligned} \tag{3.87}$$

The simulation was run for 10 sec and the simulation error is shown in 3.8. As the plot of result shows, the position error remains small on the order of 1×10^{-9} m and the pose error is smaller than the minimal threshold (1×10^{-8} rad) for the conversion algorithm to detect.

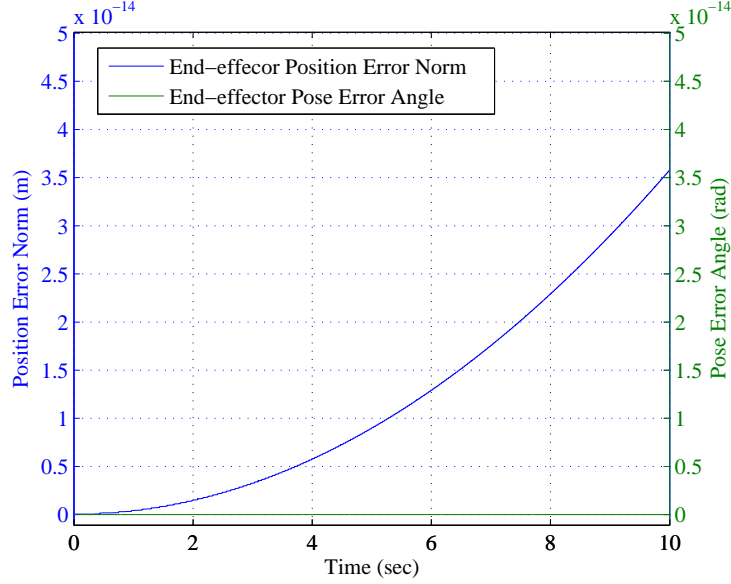


Figure 3.9: Plot of dynamics validation result with the constant test input signal.

In the third case, the input signals are kept constant. Input signals, $\tau_a(t)$, and $\tau_m(t)$ are set to zero and input $F(t) = m_T g$ [N] perfectly balances the gravity force. Thus, the center of mass of the entire system is expected remain at the same location. The manipulator link oscillates with constant amplitude like a pendulum because the simulation starts with $q = [\frac{\pi}{2}, 0]^T$ [rad], a condition away from the stable equilibrium point, and the joint natural damping is zero. The simulation was run for 10 sec and the simulation error shown in Fig. 3.9. Coordinate of the end-effector, p_E , from the MOVADYN model simulation is shown in Fig. 3.10, in which constant amplitude oscillation in both x and z coordinates is observed. The coupled movement of the VTOL in the x -direction in response to arm motions highlights the need for control of the full-order, coupled MOVA system.

From all three validation cases, simulation errors and relative simulation errors remain small, which indicates that the derived dynamic models are close to the assumed ground truth and the two derived model are close to each other. Although

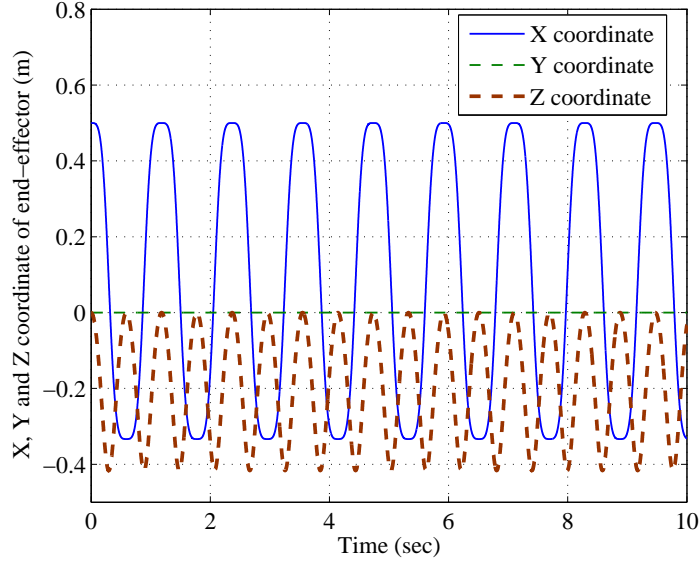


Figure 3.10: Plot of end-effector position p_E from MOVADYN derived model with the constant test input signal.

non-zero, the error is small enough to conclude that the derived models capture the system behavior and are suitable for the following model-based controller design. The constant input case also shows the system heuristically behaves as expected.

3.6 Conclusion

In summation, a framework for systematic derivation of the general MOVA system dynamics model was demonstrated. Derivation steps are presented purely with matrices and vectors to ensure conciseness of the intermediate and final results. The resulting dynamics equation is represented abstractly in the standard robotic form and is proved to have the skew-symmetric property, which is a useful property for control derivation. A program called MOVADYN was developed to achieve automatic derivation of the MOVA system dynamics for a specific aircraft and manipulator. Accessory tools are also designed to accomplish a tool chain that takes a

3D mechanical design file from Autodesk Inventor and generates code ready for use in MATLAB and Simulink for dynamics simulation or controller implementation. The dynamics equations derived using MOVADYN were validated through two different approaches. A previous manual algebraic derivation result in closed-form for a planar MOVA system was compared with the output of MOVADYN to show term-by-term correctness of the MOVADYN result. For a more complicated MOVA system in 3D space, numerical simulation of dynamics equations generated from MOVADYN and the result of numerical dynamics simulator (SimMechanics) were compared. The results suggest correctness of the dynamics equation derived from the MOVADYN program and thus the validity of the dynamics derivation approach proposed. It is believed that the systematic derivation approach presented here will facilitate the further development and control of integrated manipulator + VTOL systems.

Chapter 4

Trajectory Tracking Control of the MOVA System

4.1 Introduction

The MOVA system is a composite system of a VTOL aircraft and onboard manipulator. It is designed to be treated as a manipulator with unrestrained workspace; as such, the low-level control details should be transparent to end users who care about performing desired application tasks at an abstracted level. A trajectory tracking controller for the end-effector of the MOVA allows the end-effector to follow a reference position and orientation trajectory, predefined or generated online, and will serve as a solid inner-loop for higher-level task control. That is, the low-level control becomes a platform for further development of MOVA system application.

Previously, it was concluded that controlling the MOVA system using separately designed controllers for aircraft and manipulator will likely to result in degraded performance or even instability, which was demonstrated by previous planar MOVA test-bed experiments in Sec. 2.5.2. Thus, for the designing of this full 3D MOVA

trajectory tracking controller, the integrated control strategy promulgated in controlling, the planar MOVA system in Sec.2.4 will be pursued. However, it is not trivial to extend the controller for the planar case to a general 3D MOVA system: the dynamics of the general MOVA system have been shown to be much more intricate(Chap. 3) than that of the planar system, which is primarily attributable to increased number of degrees-of-freedom in three-dimensional system.

The mathematical model of the general MOVA system dynamics performed in Chap. 3 reveals the salient characteristics of the multi-body dynamics of the MOVA system. Albeit the dynamics are quite complex, an obvious feature is that the 3DOF dynamics of position of overall MOVA system center of mass is underactuated with one independent control input F , which represents the body-fixed thrust force vector from the rotors (or other mean of force generator) mounted on the VTOL aircraft. Linear analysis of this portion of the dynamics will indicate that the system is not controllable; however, a broader perspective suggests that the position can be controlled via rotation of the aircraft body, which re-orientes the thrust force vector. Back-stepping control design technique, as demonstrated in controllers designed for VTOL aircraft, is suitable for addressing this sort of dynamics structure [4, 29, 5]. Another property that is worth noting in the dynamics equation of the general MOVA system is that the dynamics of center of mass of the entire MOVA and the rest, which includes rotation dynamics of aircraft body and that of links of onboard manipulator, are largely decoupled due to appropriate choice of system state collection. This opens up the possibility for approaching the controller design in two steps: first applying back-stepping control design technique to the underactuated translational dynamics of MOVA system center of mass, and then control the rest of dynamics involving rotation of the VTOL aircraft and manipulator using a separate equation. Moreover, rotation in three dimensional space is a nonholonomic, and there does not exist a

trivial simplification as was used in the 2D case (where the rotation angle was described by a scalar) without introduction singularity. Such system requires special treatment in controller design since no time-invariant smooth controller is able to achieve stability (Brockett’s condition for stabilization) [45].

4.1.1 Previous Work

Back-stepping technique is a controller design method suitable for systems exhibiting strict-feedback. By systematically and recursively applying back-stepping, each layer of dynamics of a system is stabilized until the outermost layer. This technique has been successfully adopted in designing of VTOL aircraft controllers and our previous attempt for a planar MOVA system. Lee *et al.* designed an adaptive tracking controller with full state feedback for underactuated VTOL aircraft using integrator back-stepping approach and demonstrated validity of the design using both Lyapunov-type stability analysis and numerical simulation [4]. Bouabdallah *et al.* presented their results for quadrotor control with back-stepping and sliding-mode controller [29] and showed that back-stepping controller yields a more smooth control input than that from the sliding-mode controller. The previous study about the planar MOVA system end-effector tracking control demonstrated GUUB stability result with a controller designed with back-stepping technique [46].

Quaternion based controller derivation has been used in in satellite attitude control in order to avoid singularity issues inherited from Euler angle representation of the dynamics. Work of Joshi *et al.* presented a robust controller using quaternion derivation that achieves three-axis attitude stabilization of a rigid spacecraft [32]. This controller is suitable for large-angle maneuvers of satellites. It is singularity free derivation and mathematically proven to possess global asymptotic stability (GAS).

Fragopoulos and Innocenti investigated the stability issues of a quaternion based 3D attitude controller in [33], in which they adopted a discontinuous Lyapunov function to obtain GAS result for an inherently discontinuous control law. Kristiansen *et al.* also demonstrated an attitude controller using quaternion derivation for micro-satellite applications in [34, 35]. It employed an integrator backstepping derivation and obtained a controller that achieved asymptotical stability. Mayhew, Sanfelice and Teel summarized previous results in quaternion-based attitude control research and proposed a hybrid controller that introduces hysteresis to achieve robust attitude tracking [47].

4.1.2 Overview of Controller

A unified end-effector trajectory tracking controller is designed for use in the general MOVA system in three-dimensional space. GUUB stability of the proposed controller is proven using Lyapunov-type analysis. Numerical simulations of the proposed controller is also carried out to demonstrate its performance.

The control goal of achieving end-effector translational and rotational trajectory tracking is divided into tracking of the center of mass trajectory while maintaining the tracking of end-effector orientation. A trajectory of the center of mass of the entire MOVA system is generated from the reference trajectory of the end-effector.

Stabilization of the translational dynamics of the center of mass is achieved through specifying an appropriate thrust force. Since the translational dynamics are underactuated, shifting part of the actuation burden in to the rotational dynamics via backstepping control design framework. For reference trajectory tracking by the end-effector, quaternion based derivation is used in order to avoid singularity issues. Lyapunov-type stability analysis is performed on the proposed controller and Globally

Uniformly Ultimately Bounded (GUUB) stability is achieved. Simulation testing of the proposed controller is carried out in Simulink in order to demonstrate performance of the controller designed. In addition to plots of important signals, a 3D virtual reality scene is constructed to offer an animated intuitive view of system behavior.

4.2 Controller Design

This section captures the end-effector trajectory tracking controller design steps for the MOVA system in 3D space. A reference trajectory, both translational and rotational, is assumed to have adequate smoothness as derivatives of the trajectory are used in controller synthesis and implementation. The design process is performed on the 3D MOVA system with 2DOF onboard manipulator shown in Sec. 3.5.2 as this it has the minimum number of joints to achieve the trajectory tracking goal. However, the derivation could be generalized to systems with higher number of joints in the onboard manipulator. Additional joints offers opportunity for optimizing the movement of the VTOL aircraft and onboard manipulator joints under certain application specific criteria, which is left out of the discussion in this chapter.

4.2.1 Overview of Design Approach

The procedure for designing the MOVA controller is summarized as follows: first, position and orientation reference trajectories are used for calculation of a reference trajectory of the entire system center of mass using the MOVA kinematics. Then, this trajectory is used as the objective of a back-stepping controller design, which yields thrust force control input and desired angular velocity of the VTOL aircraft body. Together with the reference orientation trajectory of the end-effector, desired joint velocity is found using quaternion based rotation error system. Both the

desired aircraft body angular velocity and desired joint velocity are combined into a composite vector which is used for calculation of appropriate torque control effort based on the rotation dynamics. All resulting closed-loop dynamics are captured in a Lyapunov function which enables stability analysis of the proposed controller as a single entity. Unwanted cross terms reflected from final Lyapunov function derivative which are resulted from staged derivation are cancelled out. Note that that multi-step derivation for this controller is fundamentally different from the control strategy of separate control the VTOL aircraft and manipulator because here the derived dynamics already infer the dynamic interaction between the VTOL aircraft and the manipulator, and a controller designed based on the coupled model, though accomplished in multiple steps, is able to compensate for the interaction.

4.2.2 Generation of System Center of Mass Translational Trajectory

The reference trajectory consists of two parts, the reference trajectory of end-effector position, ${}^I p_{Er}(t)$, and that of the end-effector pose, ${}^I R_{Er}(t)$, both are assumed to be sufficiently smooth. It is also assumed that ${}^I p_{Er}(t)$ and ${}^I R_{Er}(t)$ can be used to generate a reference trajectory for the position of point C , the center of mass of the entire MOVA system, which is denoted $p_{Cr}(t)$.

For the MOVA system with two-link onboard manipulator described in Sec. 3.5.2, $p_{Cr}(t)$ can be calculated from reference trajectory ${}^I p_{Er}(t)$ and ${}^I R_{Er}(t)$ using the kinematics relationship in (3.10). First, the end-effector position

$$\begin{aligned} {}^I p_E &= {}^I p_C + {}^I R_0^0 p_{E/C} \\ &= {}^I p_C + {}^I R_E^E R_0^0 p_{E/C}, \end{aligned} \tag{4.1}$$

where

$$l_{VM} \triangleq {}^E R_0^0 p_{E/C} = {}^E p_{E/C} = \frac{(m_T + m_0)l_a}{2m_T} [0, 0, 1]^\top \quad (4.2)$$

is a constant vector. Thus, there is

$${}^I p_C = {}^I p_E - {}^I R_E l_{VM}, \quad (4.3)$$

from which the reference trajectory of point C can be calculated as

$${}^I p_{Cr}(t) = {}^I p_{Er}(t) - {}^I R_{Er}(t) l_{VM}. \quad (4.4)$$

This one-to-one mapping from ${}^I p_{Er}$ to ${}^I p_{Cr}$ with ${}^I R_{Er}$ specifically means that ${}^I p_C$ tracking a trajectory of ${}^I p_{Cr}(t)$ is equivalent to ${}^I p_E$ tracking ${}^I p_{Er}$.

4.2.3 Backstepping Control of System Center of Mass Position

Backstepping technique is utilized to facilitate control of the underactuated dynamics of position of point C, *i.e.* the system center of mass, and force ${}^I p_C(t)$ to track the reference trajectory ${}^I p_{Cr}(t)$.

The general MOVA dynamics in (3.65) is split into two equations, one describing the dynamics of p_C

$$m_t I_3 \ddot{p}_c + m_t I_3 S(\omega) \dot{p}_c + R^\top m_t g_v = F_v \quad (4.5)$$

and the other describing dynamics of ω and q

$$M_R \dot{\gamma} + C_R \gamma = \tau_{am} + B_F F_v \quad (4.6)$$

where

$$\begin{aligned}
\tau_{am} &= [\tau_a^\top, \tau_m^\top]^\top, \\
F_v &= [0 \quad 0 \quad F]^\top, \\
\gamma &\triangleq [\omega^\top, \dot{q}^\top]^\top, \\
B_F &= \begin{bmatrix} J_{\omega 0} J_{q 0} \end{bmatrix}^\top \\
M_R &= \begin{bmatrix} M_{\omega\omega} & M_{\omega q} \\ M_{\omega q} & M_{qq} \end{bmatrix}, \\
C_R &= \begin{bmatrix} C_{\omega\omega} & C_{\omega q} \\ C_{\omega q} & C_{qq} \end{bmatrix}.
\end{aligned} \tag{4.7}$$

The backstepping controller design approach will be applied to (4.5) in order to make it track the reference trajectory. First, the position error e_p is defined intuitively in frame \mathcal{B}_0 as

$$e_p \triangleq R^\top ({}^I p_C - {}^I p_{Cr}). \tag{4.8}$$

Similarly, the velocity error e_v is defined as

$$\begin{aligned}
e_v &\triangleq R^\top ({}^I \dot{p}_C - {}^I \dot{p}_{Cr}) \\
&= {}^0 \dot{p}_C - R^\top {}^I \dot{p}_{Cr}.
\end{aligned} \tag{4.9}$$

The overall filtered tracking error r is a linear combination of e_p , e_v and a constant vector δ

$$r \triangleq \alpha e_p + e_v + \delta, \tag{4.10}$$

where $\delta = [0, 0, \delta_3]^\top$ is added in the effort to connect the VTOL aircraft body rotation dynamics with the p_C dynamics and $\delta_3, \alpha \in \mathbf{R}^+$ are two control gains.

Towards building the error system dynamics, derivative is taken on e_p , e_v and r with respect to time. Time derivative of e_p is

$$\begin{aligned}\dot{e}_p &= -S(\omega)R^\top \left({}^I p_C - {}^I p_{Cr} + R^\top \left({}^I \dot{p}_C - {}^I \dot{p}_{Cr} \right) \right) \\ &= -S(\omega)e_p + e_v.\end{aligned}\tag{4.11}$$

Taking the derivative of e_v and substituting (4.5) for ${}^0\ddot{p}_c$ yields

$$\begin{aligned}\dot{e}_v &= {}^0\ddot{p}_c + S(\omega)R^\top {}^I \dot{p}_{Cr} - R^\top {}^I \ddot{p}_{Cr} \\ &= m_t^{-1}F_v - R^\top \left(g_v + {}^I \ddot{p}_{Cr} \right) - S(\omega)e_v.\end{aligned}\tag{4.12}$$

The the derivative \dot{r} is found by substituting \dot{e}_p and \dot{e}_v into $\dot{r} = \alpha\dot{e}_p + \dot{e}_v$,

$$\dot{r} = -\alpha S(\omega)e_p + \alpha e_v - S(\omega)e_v + m_t^{-1}F_v - R^\top \left(g_v + {}^I \ddot{p}_{Cr} \right).\tag{4.13}$$

From definition of r , $\alpha e_p = r - e_v - \delta$, which leads to

$$-\alpha S(\omega)e_p = -S(\omega)r + S(\omega)e_v + S(\omega)\delta.\tag{4.14}$$

Substituting (4.14) for $-\alpha S(\omega)e_p$ in (4.13) yields

$$\dot{r} = -S(\omega)r + \alpha e_v + m_t^{-1}F_v + S(\omega)\delta - R^\top \left(g_v + {}^I \ddot{p}_{Cr} \right).\tag{4.15}$$

Grouping terms that do not belong to the autonomy portion of the p_C dynamics

reveals terms that can be seen as input at this stage

$$\begin{aligned}\dot{r} &= -S(\omega)r + \alpha e_v - R^\top (g_v + {}^I\ddot{p}_{Cr}) + \begin{bmatrix} -S(\delta) & \begin{bmatrix} 0 \\ 0 \\ m_t^{-1} \end{bmatrix} \end{bmatrix} \begin{bmatrix} \omega \\ F \end{bmatrix} \\ &= -S(\omega)r + \xi_2 + B_\mu \mu,\end{aligned}\tag{4.16}$$

where $\xi_2 = \alpha e_v - R^\top (g_v + {}^I\ddot{p}_{Cr})$, intermediate input vector $\mu = [\omega^\top, F]^\top \in \mathbf{R}^4$ and intermediate input matrix $B_\mu \in \mathbf{R}^{3 \times 4}$ is defined as

$$B_\mu = \begin{bmatrix} 0 & \delta_3 & 0 & 0 \\ -\delta_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & m_t^{-1} \end{bmatrix}.\tag{4.17}$$

Since it is known that ω has its own dynamics, described by (4.6), it is not appropriate to specify μ arbitrarily as if it is a real input to the system. However, a desired intermediate input, $\mu_d = [\omega_d^\top, F_d]^\top$, that may stabilize the system can be found and the discrepancy between the desired and real intermediate input vector is defined as

$$\mu_e = \mu - \mu_d = \begin{bmatrix} \omega_e \\ F_e \end{bmatrix}.\tag{4.18}$$

The desired input μ_d is found by introducing stabilizing terms, a few terms motivated by the stability proof, and canceling terms that adversely affect of stability to yield

$$\mu_d = B_\mu^\dagger (-k_r r - \xi_2 - e_p) + (I_4 - B_\mu^\dagger B_\mu) \mu_a,\tag{4.19}$$

where $k_r = \text{diag}([k_{r1}, k_{r2}, k_{r3}])$, $k_{r1}, k_{r2}, k_{r3} \in \mathbf{R}^+$, is a control gain matrix, $I_4 \in \mathbf{R}^4$ is

an identity matrix, μ_a is an extra input term that can be freely designed in order to fit later requirements imposed during the stability analysis, and B_μ^\dagger is right pseudo inverse of B_μ , which is expanded as

$$\begin{aligned} B_\mu^\dagger &= B_\mu^\top (B_\mu B_\mu^\top)^{-1} \\ &= \begin{bmatrix} 0 & -\delta_3 & 0 & 0 \\ \delta_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & m_t \end{bmatrix}. \end{aligned} \quad (4.20)$$

Thus, the matrices $I_4 - B_\mu^\dagger B_\mu = \text{diag}([0, 0, 1, 0])$, which means $(I_4 - B_\mu^\dagger B_\mu) \mu_a = [0, 0, \mu_{a3}, 0]^\top$ assuming $\mu_a = [\mu_{a1}, \mu_{a3}, \mu_{a3}, \mu_{a4}]^\top$. Notice that this matrix maps μ_a into the nullspace of B_μ so that design of μ_a can be postponed as it will have no effect on the dynamics of r .

Substituting μ_d from (4.19) into the r -dynamics in (4.16) via $\mu = \mu_e + \mu_d$ yields the closed-loop dynamics of r ,

$$\dot{r} = -k_r r - S(\omega)r - e_p - S(\delta)\omega_e, \quad (4.21)$$

where $B_\mu \mu_e = -S(\delta)\omega_e$ because input F is a real input and $F \equiv F_d$ which leads to $F_e \equiv 0$. The term $S(\delta)\omega_e$ is regarded as a disturbance to this part of the system dynamics and $|S(\delta)\omega_e|$ should be minimized in a secondary controller.

4.2.4 Control of Rotation Dynamics of Aircraft and Manipulator

Dynamics of state vector γ in (4.6) describes composite rotation dynamics of the VTOL aircraft and that of the onboard manipulator. In (4.19), the desired VTOL

aircraft body angular velocity is specified as a part of μ_d . The desired joint velocity \dot{q}_d is assumed to be known at this point in derivation, the generation of \dot{q}_d from system state and reference trajectory $R_{Er}(t)$ is covered in next section.

The joint velocity error is defined as

$$\dot{q}_e \triangleq \dot{q} - \dot{q}_d. \quad (4.22)$$

Concatenating \dot{q}_d and ω_d , $\gamma_d \in \mathbf{R}^{3+N}$ is defined as

$$\gamma_d \triangleq \begin{bmatrix} \omega_d \\ \dot{q}_d \end{bmatrix}. \quad (4.23)$$

The error of the composite vector, γ_e , is thus

$$\gamma_e = \gamma - \gamma_d = \begin{bmatrix} \omega_e \\ \dot{q}_e \end{bmatrix}, \quad (4.24)$$

which provides another representation of γ in the form

$$\gamma = \gamma_e + \gamma_d. \quad (4.25)$$

Substituting (4.25) into (4.6), yields

$$M_R \dot{\gamma}_e + M_R \dot{\gamma}_d + C_R \gamma = \tau_{am} + B_F F_v, \quad (4.26)$$

which is reorganized into

$$\begin{aligned} M_R \dot{\gamma}_e &= \tau_{am} + B_F F_v - C_R \gamma - M_R \dot{\gamma}_d \\ &= \tau_{am} - \xi_3, \end{aligned} \tag{4.27}$$

where short-hand notation $\xi_3 = -B_F F_v + C_R \gamma + M_R \dot{\gamma}_d$ was introduced.

Control of dynamics of error γ_e in (4.27) is designed for τ_{am} as

$$\tau_{am} = -k_\gamma \gamma_e + \xi_3 + \begin{bmatrix} S(\delta)r \\ 0 \end{bmatrix} + \tau_c, \tag{4.28}$$

where $k_\gamma = \text{diag}([k_{\gamma 1}, k_{\gamma 2}, k_{\gamma 3}, k_{\gamma 4}, k_{\gamma 5}])$, $(k_{\gamma i} \in \mathbf{R}^+, 1 \leq i \leq 5)$, is a control gain matrix and $\tau_c \in \mathbf{R}^5$ is a cross term added based on the ensuing stability proof after joint velocity vector vector is introduced. The close-loop dynamics of γ_e is thus

$$M_R \dot{\gamma}_e = -k_\gamma \gamma_e - C_R \gamma_e + \begin{bmatrix} S(\delta)r \\ 0 \end{bmatrix} + \tau_c. \tag{4.29}$$

4.2.5 Desired Joint Velocity of End-effector

Explicit description of the desired joint velocity \dot{q}_d is so far missing in the development of the rotation dynamics control law, it will be provided in this section. The desired joint velocity developed using a quaternion representation of rotation in order to avoid singularities in the controller. To prevent overloading the symbol q , which has already been assigned to the onboard manipulator joint configuration vector, Q is used to denote a quaternion (rather than following the more popular choice q). A quaternion $Q \in \mathbf{R}^4$ can be separated into a scalar part η and a vector

part $\epsilon \in \mathbf{R}^3$:

$$Q \triangleq \begin{bmatrix} \eta \\ \epsilon \end{bmatrix}^\top. \quad (4.30)$$

By definition, $\|Q\| = 1$ and $Q \equiv -Q$ when Q is used to represent a rotation in three-dimensional space. Disregarding the sign ambiguity, it is easy to form a mapping between a rotation matrix R , also called the directional cosine matrix, to Q . To maintain compatibility with the previous dynamics derivation for the MOVA system that uses the rotation matrix, a quaternion Q will be denoted using the same decoration system, i.e. rotation matrix bR_a is equivalent to quaternion bQ_a , in terms of the rotation they denote, where a is the placeholder for the frame of origination and b is the destination. Rotation dynamics described using rotation matrix is

$$\frac{d}{dt} {}^bR_a = S({}^b\omega_{a/b}) {}^bR_a \quad (4.31)$$

which represented using quaternion is

$$\frac{d}{dt} {}^aQ_b = \begin{bmatrix} \dot{\eta} \\ \dot{\epsilon} \end{bmatrix} = \begin{bmatrix} -\frac{1}{2}\epsilon^\top {}^a\omega_{a/b} \\ \frac{1}{2}[\eta I_3 + S(\epsilon)] {}^a\omega_{a/b} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} -\epsilon^\top \\ \eta I_3 + S(\epsilon) \end{bmatrix} {}^b\omega_{a/b}, \quad (4.32)$$

while ${}^aQ_b = [\eta, \epsilon^\top]^\top$.

Let a rotation matrix R denote the current orientation and R_r to denote the reference orientation, and the error between the two, R_{err} , satisfies $R_r R_{err} = R$, which leads to $R_{err} = R_r^\top R$. In the quaternion representation, where Q is equivalent to R and Q_r is equivalent to R_r , the error \tilde{Q} is

$$\tilde{Q} = \bar{Q}_r \otimes Q, \quad (4.33)$$

where $\bar{Q} = [\eta, -\epsilon^\top]^\top$ is the quaternion conjugate of $Q = [\eta, \epsilon^\top]^\top$ (note that $\bar{Q} \otimes Q = Q \otimes \bar{Q} = Q_I$, $Q_I = [1, 0, 0, 0]^\top$ is the identity quaternion), and the operator \otimes denotes a quaternion product. Expanding the quaternion product, \tilde{Q} is

$$\tilde{Q} = \begin{bmatrix} \tilde{\eta} \\ \tilde{\epsilon} \end{bmatrix} = \begin{bmatrix} \eta_r \eta + \epsilon_r^\top \eta \\ \eta \epsilon_r + \eta_r \epsilon - S(\epsilon_r) \epsilon \end{bmatrix}, \quad (4.34)$$

where $Q_r = [\eta_r, \epsilon_r^\top]^\top$. Further reorganization of terms in (4.34) can be done to yield representations of \tilde{Q} that are linear with respect to either Q or Q_r

$$\tilde{Q} = \begin{bmatrix} \eta_r & \epsilon_r^\top \\ -\epsilon_r & \eta_r I_3 - S(\eta_r) \end{bmatrix} Q = \begin{bmatrix} \eta & \epsilon^\top \\ \epsilon & -\eta I_3 + S(\epsilon) \end{bmatrix} Q_r \quad (4.35)$$

The time derivative of \tilde{Q} is obtained using the chain-rule,

$$\begin{aligned} \frac{d}{dt} \tilde{Q} &= \frac{\partial \tilde{Q}}{\partial Q} \frac{d}{dt} Q + \frac{\partial \tilde{Q}}{\partial Q_r} \frac{d}{dt} Q_r \\ &= \begin{bmatrix} \eta_r & \epsilon_r^\top \\ -\epsilon_r & \eta_r I_3 - S(\eta_r) \end{bmatrix} \frac{d}{dt} Q + \begin{bmatrix} \eta & \epsilon^\top \\ \epsilon & -\eta I_3 + S(\epsilon) \end{bmatrix} \frac{d}{dt} Q_r. \end{aligned} \quad (4.36)$$

If quaternions Q , Q_r represents actual or reference orientation trajectory of an object, they conform to dynamics equations

$$\frac{d}{dt} Q = \begin{bmatrix} \dot{\eta} \\ \dot{\epsilon} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} -\epsilon^\top \\ \eta I_3 + S(\epsilon) \end{bmatrix} \omega \quad (4.37)$$

and

$$\frac{d}{dt} Q_r = \begin{bmatrix} \dot{\eta}_r \\ \dot{\epsilon}_r \end{bmatrix} = \frac{1}{2} \begin{bmatrix} -\epsilon_r^\top \\ \eta_r I_3 + S(\epsilon_r) \end{bmatrix} \omega_r. \quad (4.38)$$

Substituting (4.37) and (4.38) into (4.36) yields

$$\dot{\tilde{Q}} = \frac{1}{2} \begin{bmatrix} -\tilde{\epsilon}^\top \\ \tilde{\eta}I_3 + S(\tilde{\epsilon}) \end{bmatrix} (\omega - \omega_r) + \begin{bmatrix} 0 \\ S(\tilde{\epsilon}) \end{bmatrix} \omega_r. \quad (4.39)$$

An error system is to be designed to reach zero when \tilde{Q} reaches identity, which is equivalent to $R_{err} = I_3$. Notice $Q \equiv -Q$, so that reaching identity means $\tilde{Q} = [1, 0, 0, 0]^\top$ or $\tilde{Q} = [-1, 0, 0, 0]^\top$. To satisfy this requirement, attitude error z_1 is defined as a function of \tilde{Q}

$$z_1 = \begin{bmatrix} 1 - |\tilde{\eta}| \\ \tilde{\epsilon} \end{bmatrix}. \quad (4.40)$$

The time derivative of z_1 can be obtained as a function of \tilde{Q} and $\dot{\tilde{Q}}$

$$\dot{z}_1 = \begin{bmatrix} -\text{sgn}(\tilde{\eta})\dot{\tilde{\eta}} \\ \dot{\tilde{\epsilon}} \end{bmatrix} = \begin{bmatrix} -\text{sgn}(\tilde{\eta}) & 0 \\ 0 & I_3 \end{bmatrix} \dot{\tilde{Q}}. \quad (4.41)$$

Substituting (4.39) into \dot{z}_1 , yields

$$\begin{aligned} \dot{z}_1 &= \frac{1}{2} \begin{bmatrix} \text{sgn}(\tilde{\eta})\tilde{\epsilon}^\top \\ \tilde{\eta}I_3 + S(\tilde{\epsilon}) \end{bmatrix} (\omega - \omega_r) + \begin{bmatrix} 0 \\ S(\tilde{\epsilon}) \end{bmatrix} \omega_r \\ &= \frac{1}{2} G^\top \omega + \left(\begin{bmatrix} 0 \\ S(\tilde{\epsilon}) \end{bmatrix} - \frac{1}{2} G^\top \right) \omega_r, \end{aligned} \quad (4.42)$$

where

$$G = \begin{bmatrix} \text{sgn}(\tilde{\eta})\tilde{\epsilon}^\top \\ \tilde{\eta}I_3 + S(\tilde{\epsilon}) \end{bmatrix}^\top. \quad (4.43)$$

To facilitate for orientation tracking, it will be desirable to have $z_1 = 0$ initially

and then ω follows ω_r exactly to maintain $z_1 = 0$. However, these assumptions about both the initial condition and the ability to maintain $\omega_r = \omega$ are not realistic. Even if ω tracks ω_r , but z_1 is not zero, the error will at least be maintained. If z_1 is not zero, continue to let A desired angular velocity ω_d is designed to shrink the attitude zero

$$\omega_d = -K_z G z_1 + \omega_r, \quad (4.44)$$

where $K_z = [k_{z1}, k_{z2}, k_{z3}] (k_{z1}, k_{z2}, k_{z3} \in \mathbf{R}^+)$ is a control gain matrix. Substituting this control law into (4.41), yields the close-loop dynamics of z_1

$$\dot{z}_1 = -\frac{1}{2} G^\top K_z G z_1 - \frac{1}{2} G^\top \omega_e + \begin{bmatrix} 0 \\ S(\tilde{\epsilon}) \end{bmatrix} \omega_r, \quad (4.45)$$

where intermediate error $\omega_e = \omega_d - \omega$ was introduced.

In the case of generating a desired angular velocity for the end-effector, Q represents the actual orientation of end-effector, Q_r represents the reference orientation of end-effector, which is specified as part of the reference trajectory. The term ω will use end-effector angular velocity ${}^E\omega_E$ in the end-effector fixed frame, which can be found using kinematics in Sec. 3.2 as

$${}^E\omega_E = {}^E R_0 {}^0\omega_E = {}^E R_0 (\omega + J_{\omega E} \dot{q}). \quad (4.46)$$

The term ω_r is the reference angular velocity of the end-effector in the end-effector fixed frame that satisfies (4.36). The resulting ω_d from (4.44) indicate the desired end-effector angular velocity in the end-effector fixed frame, or ${}^E\omega_{Ed}$, which satisfies

$${}^0 R_E {}^E\omega_{Ed} = {}^0\omega_{Ed} = \omega_d + J_{\omega E} \dot{q}_d = \begin{bmatrix} I_3 & J_{\omega E} \end{bmatrix} \dot{\gamma}_d. \quad (4.47)$$

From the definition of γ_d in (4.23),

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \gamma_d = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \mu_d|_{\mu_a=0}. \quad (4.48)$$

Together, it is possible to form an linear equation to solve for γ_d , which is assumed to be known in Sec. 3.2

$$\begin{aligned} \gamma_d &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ & & I_3 & J_{\omega E} \end{bmatrix}^{-1} \begin{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \mu_d|_{\mu_a=0} \\ {}^0 R_E^E \omega_{Ed} \end{bmatrix} \\ &= \Gamma_{d1}^{-1} \Gamma_{d2}, \end{aligned} \quad (4.49)$$

where Γ_{d1} and Γ_{d2} are short-hand notations for the two matrices on the right-hand side of (4.49), from left to right, respectively. The Γ_{d1} is invertible if $J_{\omega E}$ is full rank. For the MOVA system described in Sec. 3.5.2, this is obtained if $q_1 \neq 2k\pi \pm \pi/2$.

The auxiliary input μ_a is derived from (4.49) and (4.19)

$$\mu_a = [0, 0, \mu_{a3}, 0]^T = \text{diag}([0, 0, 1, 0, 0]) \gamma_d - \text{diag}([0, 0, 1, 0]) \mu_d|_{\mu_a=0}. \quad (4.50)$$

The additional input τ_c in (4.28) is designed to be

$$\tau_c = \begin{bmatrix} Gz_1 \\ 0 \end{bmatrix}, \quad (4.51)$$

based on the role of z_1 in the stability analysis.

4.3 Stability Analysis

In this section, stability of the proposed controller is analyzed with a Lyapunov-type approach. The controller is proven to provide Globally Uniformly Ultimately Bounded (GUUB) tracking of the desired position and orientation trajectories.

A positive definite, radially unbounded function V is proposed for the stability analysis as

$$V = \frac{1}{2}e_p^\top e_p + \frac{1}{2}r^\top r + \frac{1}{2}\gamma_e^\top M_R \gamma_e + \frac{1}{2}z_1^\top z_1. \quad (4.52)$$

The time derivative of (4.52) is

$$\dot{V} = e_p^\top \dot{e}_p + r^\top \dot{r} + \gamma_e^\top M_R \dot{\gamma}_e + \frac{1}{2}\gamma_e^\top \frac{d}{dt}(M_R)\gamma_e + z_1^\top \dot{z}_1. \quad (4.53)$$

Substituting the dynamics of e_p from (4.11) and the closed-loop dynamics of r , γ_e and z_1 in (4.21), (4.29), (4.45) into (4.53), yields

$$\begin{aligned} \dot{V} = & e_p^\top (-S(\omega)e_p + e_v) + \\ & r^\top (-S(\omega)r - k_r r - e_p - S(\delta)\omega_e) + \\ & \gamma_e^\top \left(-k_\gamma \gamma_e - C_R \gamma_e + \begin{bmatrix} S(\delta)r + Gz_1 \\ 0 \end{bmatrix} \right) + \frac{1}{2}\gamma_e^\top \frac{d}{dt}(M_R)\gamma_e + \\ & z_1^\top (G^\top k_1 G z_1 - G\omega_e). \end{aligned} \quad (4.54)$$

Expanding all terms, \dot{V} can be rewritten as

$$\begin{aligned} \dot{V} = & e_p^\top e_v - r^\top k_r r - r^\top e_p - r^\top S(\delta)\omega_e - \gamma_e^\top k_\gamma \gamma_e + \omega_e^\top G^\top z_1 + \omega_e^\top S(\delta)\gamma_e \\ & - \gamma_e^\top C_R \gamma_e + \frac{1}{2}\gamma_e^\top \frac{d}{dt}(M_R)\gamma_e - z_1^\top G^\top k_z G z_1 - z_1^\top G\omega_e, \end{aligned} \quad (4.55)$$

where skew-symmetric matrix property of $S(\omega)$ is invoked to get $e_p^\top S(\omega)e_p = 0$ and

$r^\top S(\omega)r = 0$ and

$$\gamma_e^\top \begin{bmatrix} S(\delta)r + Gz_1 \\ 0 \end{bmatrix} = \omega_e^\top G^\top z_1 + \omega_e^\top S(\delta)\gamma_e. \quad (4.56)$$

Terms that cancel each other in (4.55) include $-r^\top S(\delta)\omega_e + \omega_e^\top S(\delta)\gamma_e = 0$, $\omega_e^\top G^\top z_1 - z_1^\top G\omega_e = 0$ which is a direct result from the fact that they are scalars; moreover, there is $e_p^\top e_v - r^\top e_p = -\alpha e_p^\top e_p - e_p^\top \delta$ from (2.37). Applying these identities into (4.55), yields

$$\dot{V} = -\alpha e_p^\top e_p - r^\top k_r r - \gamma_e^\top k_\gamma \gamma_e - z_1^\top G^\top k_z G z_1 - e_p^\top \delta + \gamma_e^\top \left(\frac{1}{2} \frac{d}{dt} (M_R) - C_R \right) \gamma_e. \quad (4.57)$$

Moreover, according to Sec. 3.3.6, $\frac{1}{2} \frac{d}{dt} M - C$ is skew-symmetric. It then follows that a submatrix symmetric to the diagonal, $\frac{1}{2} \frac{d}{dt} M_R - C_R$, is also skew-symmetric. Thus, $\gamma_e^\top \left(\frac{1}{2} \frac{d}{dt} (M_R) - C_R \right) \gamma_e = 0$ and

$$\dot{V} = -\alpha e_p^\top e_p - r^\top k_r r - \gamma_e^\top k_\gamma \gamma_e - z_1^\top G^\top K_z G z_1 - e_p^\top \delta, \quad (4.58)$$

of which first four quadratic terms are identified as stabilizing terms for the error dynamics.

The second last term in (4.58) is expanded as

$$\begin{aligned} z_1^\top G^\top K_z G z_1 &= k_z (G z_1)^\top G z_1 \\ &= k_z \tilde{\epsilon}^\top \tilde{\epsilon} \\ &= k_z (1 - \tilde{\eta}^2) \end{aligned} \quad (4.59)$$

since

$$Gz_1 = \begin{bmatrix} \text{sgn}(\tilde{\eta})\tilde{\epsilon} & \tilde{\eta}I_3 + S(\tilde{\epsilon}) \end{bmatrix} \begin{bmatrix} 1 - \tilde{\eta} \\ \tilde{\epsilon} \end{bmatrix} = \text{sgn}(\tilde{\eta})\tilde{\epsilon}. \quad (4.60)$$

By definition of z_1 in (4.40),

$$\begin{aligned} z_1^\top z_1 &= 1 - 2|\tilde{\eta}| + |\eta|^2 + \tilde{\epsilon}^\top \tilde{\epsilon} \\ &= 2 - 2|\tilde{\eta}|, \end{aligned} \quad (4.61)$$

because $|\eta|^2 + \tilde{\epsilon}^\top \tilde{\epsilon} = 1$. As $|\tilde{\eta}| \in [0, 1]$, there is

$$\frac{1}{2}k_z z_1^\top z_1 \leq z_1^\top G^\top K_z G z_1 \leq k_z z_1^\top z_1 \quad (4.62)$$

The upper bound of the last term $-e_p^\top \delta$ in (4.58) is found using Cauchy-Schwarz inequality

$$\begin{aligned} -e_p^\top \delta &\leq \|e_p\| \cdot \|\delta\| \\ &\leq \frac{1}{2} \left(\lambda_1 \|e_p\|^2 + \frac{1}{\lambda_1} \delta_3^2 \right), \end{aligned} \quad (4.63)$$

where $\lambda_1 \in \mathbf{R}^+$. Therefore,

$$\begin{aligned} \dot{V} &\leq - \left(\alpha - \frac{\lambda_1}{2} \right) e_p^\top e_p - r^\top k_r r - \gamma_e^\top k_\gamma \gamma_e - z_1^\top G^\top k_z G z_1 + \frac{1}{2\lambda_1} \delta_3^2 \\ &\leq -2\lambda_2 V + \frac{1}{2\lambda_1} \delta_3^2, \end{aligned} \quad (4.64)$$

where a scalar $\lambda_2 \in \mathbf{R}^+$ is given by

$$\lambda_2 = \min \left\{ \alpha - \frac{\lambda_1}{2}, k_{ri}, \frac{k_{\gamma i}}{\lambda_{\max}(M_R)}, \frac{k_z}{2} \right\}, \quad (4.65)$$

where $\lambda_{\max}(\cdot)$ denotes the largest eigenvalue of the matrix. Solving the differential

inequality in (4.64) yields

$$V \leq V_0 \cdot e^{-2\lambda_2 t} + \frac{\delta_3^2}{4\lambda_2\lambda_1}(1 - e^{-2\lambda_2 t}), \quad (4.66)$$

where V_0 is V evaluated at the initial condition ($t = 0$).

A complete tracking error vector of the control system is defined as

$$\Omega = \left[e_p^\top, r^\top, \sqrt{M_R}\gamma_e, z_1 \right]^\top, \quad (4.67)$$

which can be rewritten as

$$\frac{1}{2} \|\Omega\|^2 = V. \quad (4.68)$$

Substituting (4.68) into (4.66), multiplying by 2 and taking square root on both sides yields,

$$\|\Omega(t)\| \leq \sqrt{\|\Omega_0\|^2 e^{-2\lambda_2 t} + \frac{1}{2\lambda_2\lambda_1} \delta_2^2 (1 - e^{-2\lambda_2 t})}, \quad (4.69)$$

where Ω_0 is η evaluated at $t = 0$. This result represents the steady state bound on $\|\Omega(t)\|$ as

$$\lim_{t \rightarrow \infty} \|\Omega(t)\| = \frac{\delta_2}{\sqrt{2\lambda_1\lambda_2}}. \quad (4.70)$$

In other words, the norm of the complete tracking error, $\eta(t)$, is Globally Uniformly Ultimately Bounded (GUUB). From this results, all signals in this controller can be shown bounded via signal tracing.

4.4 Simulation

Simulation of the proposed controller is shown in this section in an effort to demonstrate controller validity. The 3D MOVA system with two-link onboard ma-

nipulator (described in Sec. 3.5.2) is targeted for the simulation. Physical parameters of the MOVA system remain as the same as in Sec. 3.5.2, which highlights the low mass ratio between VTOL aircraft body and the manipulator link, a condition that leads to deteriorated operation of a controller that uses “separate” control strategy for the manipulator and aircraft. The goal of the simulation is to demonstrate that the proposed controller achieves the objective of directing the end-effector of this MOVA system to follow predefined position and orientation trajectory. Multiple test reference trajectories with distinct characteristics are used to show the validity of the controller under different conditions. Results of the simulation are shown as error curves for analysis and also displayed in real simulation time in the form of synthetic animated graphics for better illustration of system behavior in realtime (see Fig. 4.1). The background scenery and a mock up of the MOVA system are specified using VRML (Virtual Reality Modeling Language) using the Simulink Virtual Reality Toolbox . Simulation signals are fed into the VRML viewer during runtime so that the graphics rendered represent system motion.

Position trajectory of the center of mass of the overall MOVA system has to be evaluated from composite position and orientation trajectory of the end-effector using Equation (4.4). Both the resulting position trajectory of the center of mass and the orientation trajectory for the end-effector are input into the controller as required. In all simulations the end-effector operates above the aircraft body. The initial condition for all testing is $p_C = [0, 0, 0]^T \text{m}$, $R = I_3$ and $q = [0, 0]^T \text{rad}$.

The first test reference trajectory is movement from the origin to the single set point specified as $p_E = [5, 5, 3]^T \text{m}$ and $R_E = R_{zzx}(\frac{\pi}{4}, \frac{\pi}{8}, \frac{\pi}{8})$, where R_{zzx} denotes resulting rotation matrix associated with Euler angle using the common z-x-z convention. Thus, R_E is obtained from three successive rotations in the body frame: first, around z-axis by $\frac{\pi}{4} \text{rad/s}$; then, around x-axis by $\frac{\pi}{8} \text{rad/s}$ and finally, around the z-axis by $\frac{\pi}{8}$

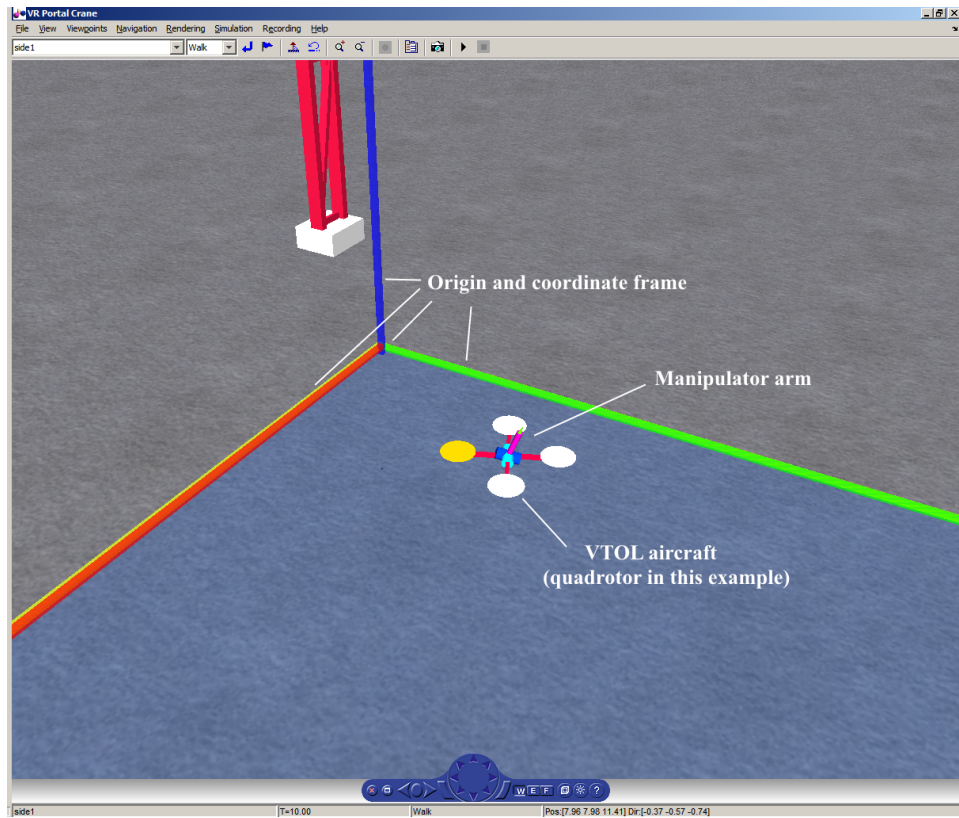


Figure 4.1: Screenshot of the VRML viewer during simulation.

rad/s.

The set point equals a step input to the controller. In order to conform to the assumption of smoothness of the reference trajectory, set point of both position and orientation of end-effector is low-pass filtered to achieve adequate smoothness. For the position reference trajectory, the coordinates of set point are filtered with

$$T_{lp}(s) = \frac{4}{s^2 + 4s + 4}, \quad (4.71)$$

which is a critically-damped low-pass filter with natural frequency 2 rad/s. For the orientation reference trajectory, the set point is converted into axis-angle representation and the amount of angle is low-pass filtered using the same filter as (4.71).

The result is shown in terms of error in end-effector position (see Fig. 4.2) and orientation (see Fig. 4.3) . The end-effector position error is measured as the distance between the reference and actual position (i.e. the 2-norm) while the orientation error is measured as the minimum amount of rotation to gap the discrepancy between the reference and actual orientation. At steady-state, the position error is about 0.002 m and the orientation error is less than 0.002 rad.

To visualize how the actual trajectory converges to the reference, a three dimensional plot is displayed in Fig. 4.4 where both the reference trajectory of the end-effector and the calculated reference trajectory of the center of mass (CM) of the aircraft are shown along with the actual trajectories.

The second reference trajectory commands the end-effector position to go through a composite curve, while the end-effector orientation trajectory enables the tip of the end-effector to point to a fixed point in the inertial frame. A visualization

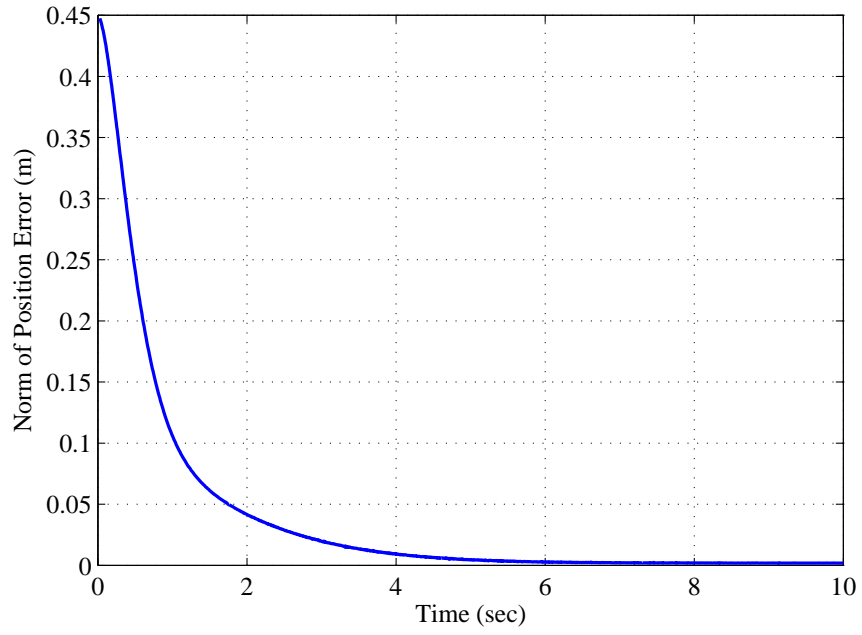


Figure 4.2: Plot of of position error of end-effector norm.

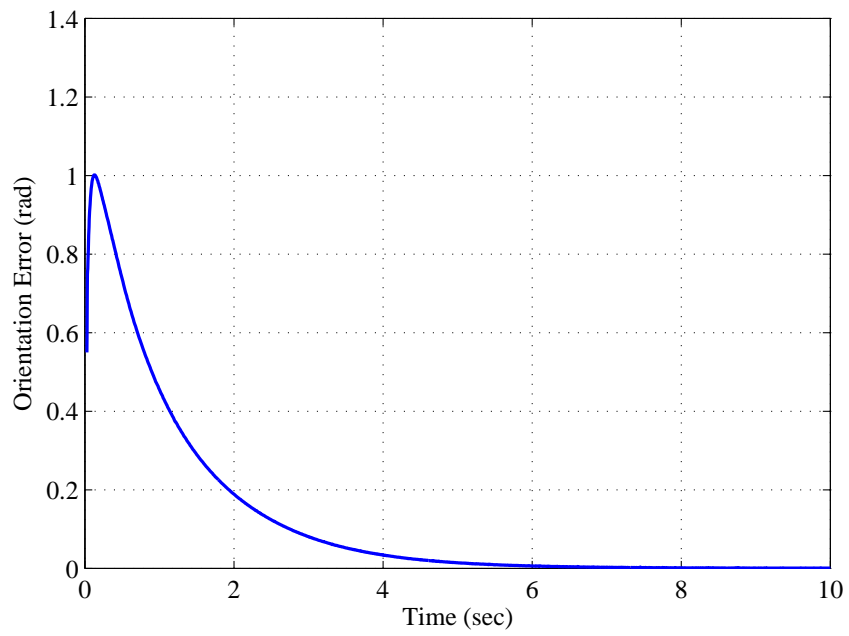


Figure 4.3: Plot of orientation error of end-effector in terms of minimum rotation angle.

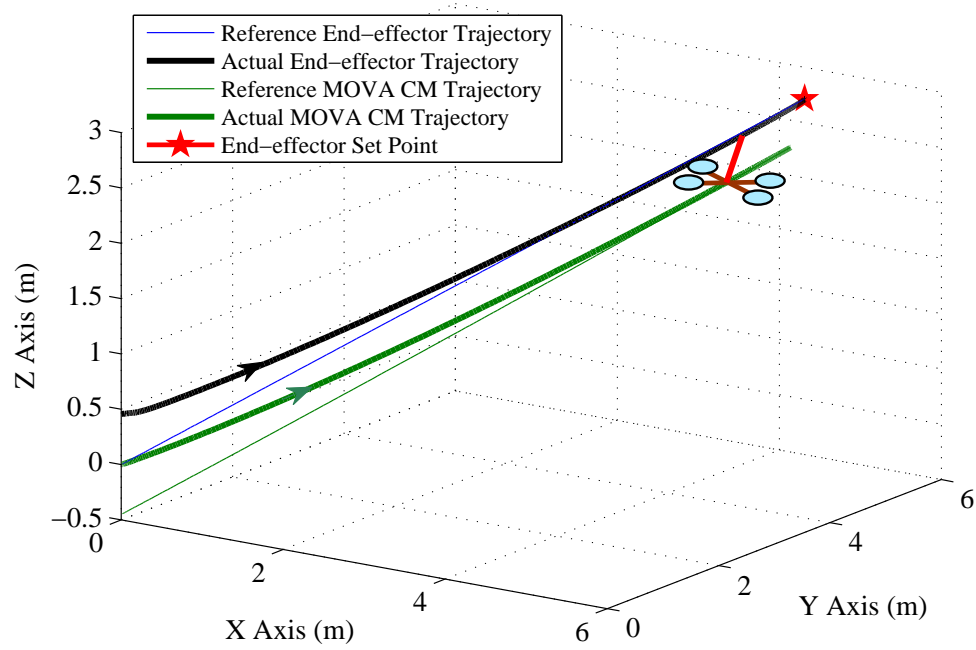


Figure 4.4: Spatial visualization of actual and reference trajectory of both end-effector and CM of entire MOVA system.

of the trajectory is shown in Fig. 4.5. The trajectory for the end-effector position is

$${}^I p_E = \begin{bmatrix} A_{xy} \cos(\omega_{xy}t) \\ A_{xy} \sin(\omega_{xy}t) \\ z_0 + A_z \sin(\omega_z t) \end{bmatrix} [\text{m}], \quad (4.72)$$

where t denotes simulation time, $A_{xy} = 5$ m, $\omega_{xy} = 0.2\pi$ rad/s, $z_0 = 2$ m, $A_z = 0.2$ m and $\omega_z = 1.6\pi$ rad/s. Projecting the trajectory to x-y plane in the inertial frame, it forms a circle of radius A_{xy} ; at the time, the motion in the z -axis direction is sinusoidal with amplitude of A_z . The trajectory can be seen as a composite motion of going along circle horizontally while moving up and down vertically. The orientation of the end-effector is derived from the position trajectory and a fixed point $p_F = [0, 0, 5]^T$ m in the inertial frame so that the end-effector always points to p_F . A body-frame affixed to the end-effector, with the z -axis pointing outward along the long direction of the manipulator link, constrains the y -axis of the frame to remain parallel to the x - y plane of the inertial frame at all time. This complex reference trajectory is visualized in Fig. 4.5.

The result is demonstrated by plots of end-effector position and orientation error in Fig.4.6 and 4.7, respectively. At steady-state, the position error is less than 0.02 m and the orientation error is less than 0.02 rad. Visualization of the actual and reference trajectory of the end-effector is provided in Fig. 4.8.

For the third reference trajectory, the end-effector position remains at a fixed point, while the end-effector orientation rotates back and forth. The position and orientation trajectory of the end-effector are expressed by $p_E = [5, 5, 5]^T$ and $R_E = R_{zz}(0, A_x \sin(\omega_x t))$, where $A_x = 1$ rad and $\omega_x = 1$ rad/s. This motion mimics the physical movement of rotating a ratchet wrench for tightening or loosening a bolt

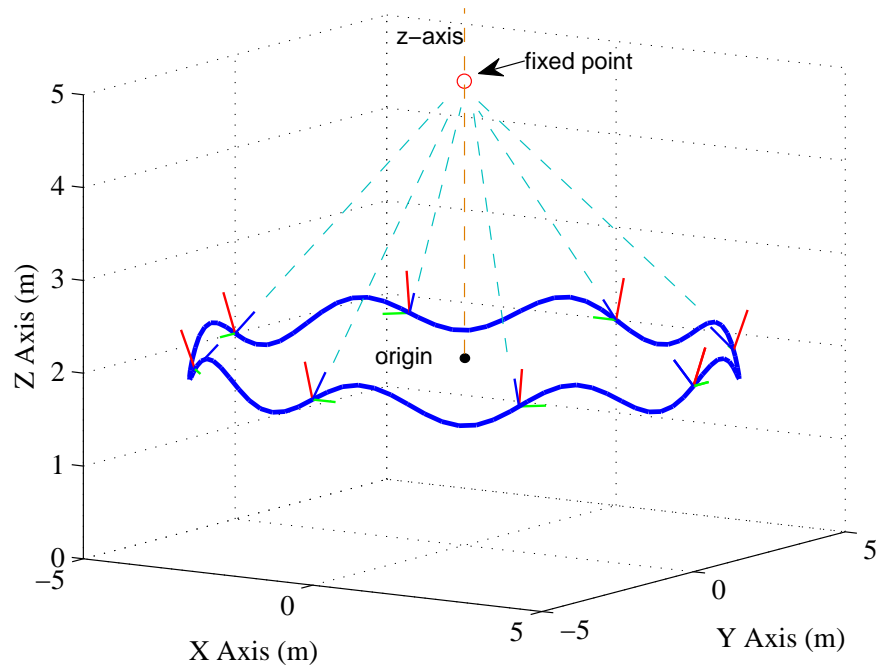


Figure 4.5: Visualization of the composite position and orientation trajectory. The thick blue curve is the position trajectory of the end-effector. The orientation trajectory is marked out sparsely with small frame icons attached to the position trajectory. The blue line the is z -axis, and green line is the y -axis. the z -axis is always directed to the fixed point

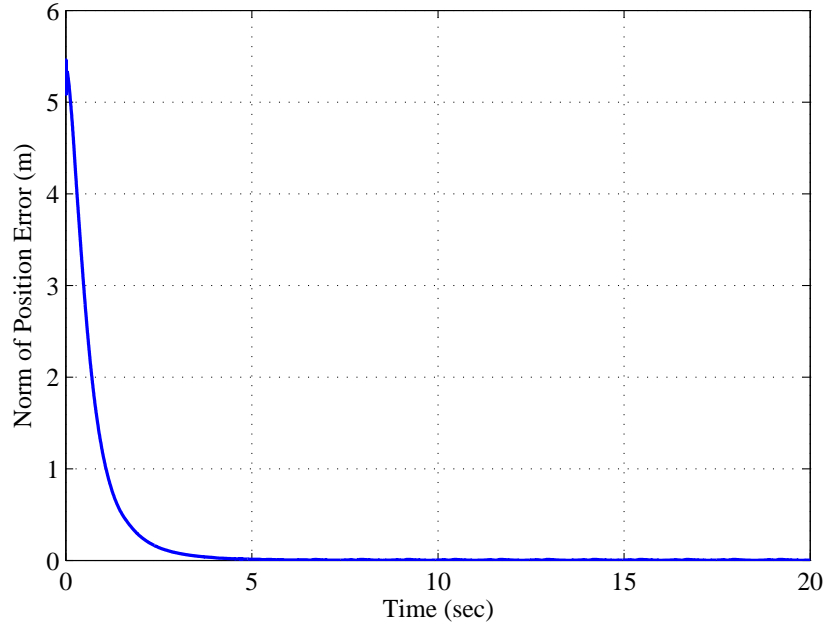


Figure 4.6: Plot of position error of end-effector in term of distance.

(although the force and torque interaction with a real bolt is not considered in this case).

Position and orientation error of the end-effector of the system is plotted in Fig. 4.9 and Fig. 4.9, respectively. At steady-state, the position error is less than 0.003 m and the orientation error is less than 0.004 rad.

Simulation results of the proposed controller on a 3D MOVA system with two-joint onboard manipulator was demonstrated. Under three reference trajectory with distinct characteristics, the controller was able to regulate the error in the end-effector position and orientation in a short period of time and maintain the error at low steady-state level afterwards, which suggests that the proposed trajectory tracking controller is effective.

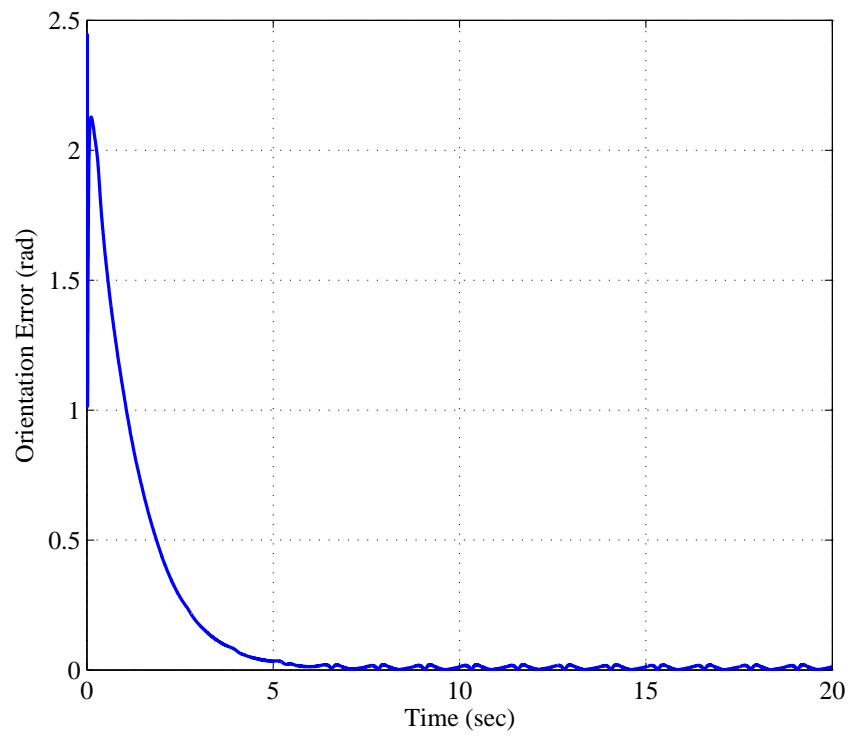


Figure 4.7: Plot of orientation error of end-effector in term of minimum rotation angle.

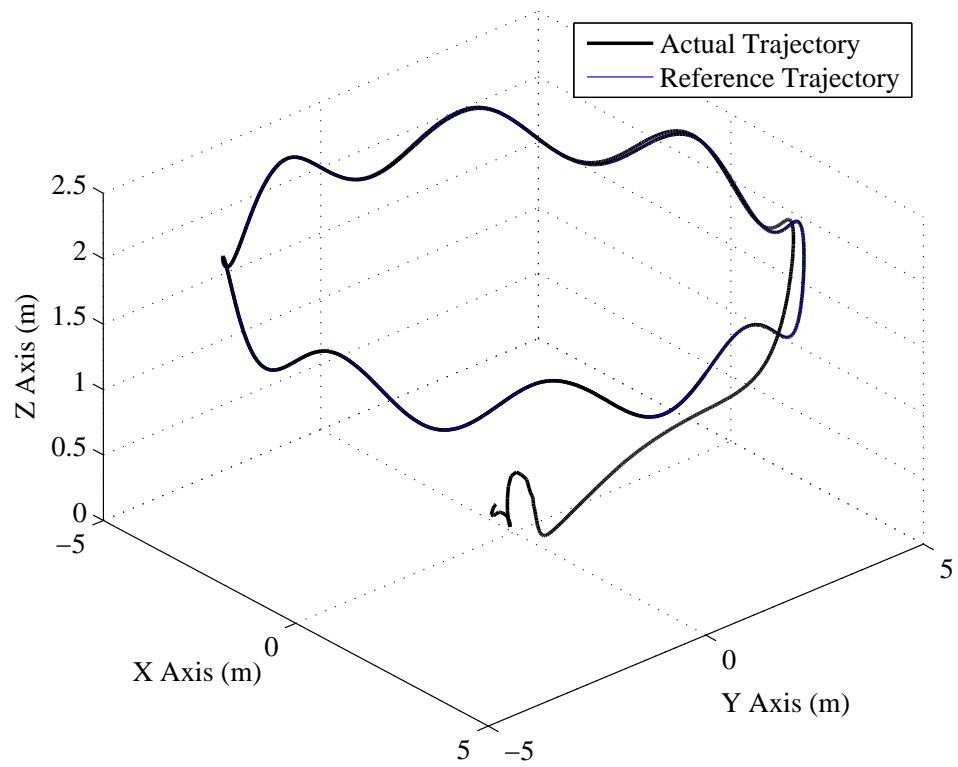


Figure 4.8: Visualization of the actual and reference trajectory of end-effector position in 3D space.

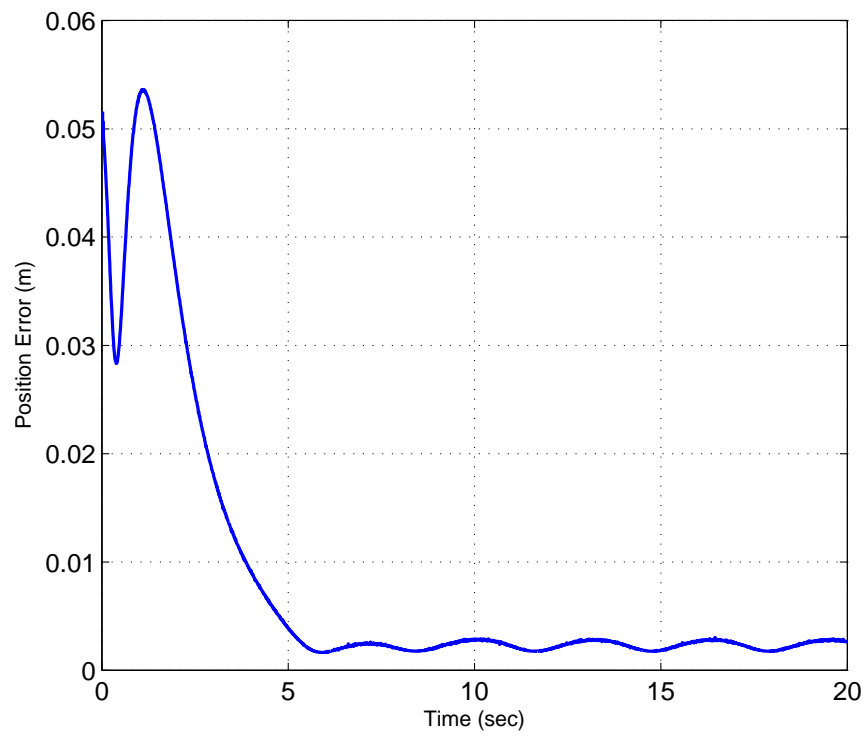


Figure 4.9: Plot of position error of end-effector in term of distance.

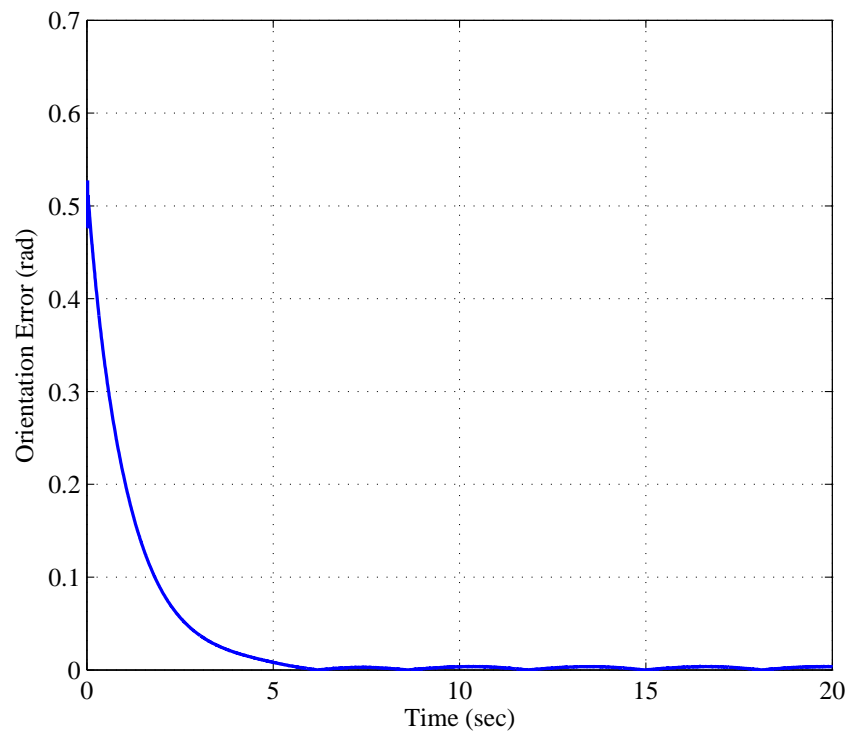


Figure 4.10: Plot of orientation error of end-effector in term of minimum rotation angle.

4.5 Conclusion

In summation, a unified nonlinear control algorithm that controls the MOVA system, including both the aircraft and the onboard manipulator, as single entity is developed to achieve trajectory tracking of the MOVA end-effector position and attitude. Reference trajectory of the end-effector is processed to yield equivalent reference positional trajectory for system center of mass. Backstepping technique is applied to connect the rotation dynamics of aircraft to the underactuated translational dynamics of the overall system center of mass. End-effector orientation trajectory is combined with attitude of aircraft in runtime to generate manipulator reference trajectory in joint space, which is used in control effort calculation for joint torques.

The stability of proposed controller was proven using Lyapunov-type stability analysis which resulted in Globally Uniformly Ultimately Bounded (GUUB) of the states in the analysis function. All signals are shown bounded afterwards from this result.

Simulation test-bed is also constructed to evaluate performance of the proposed controller on a MOVA system with a two-link onboard manipulator. The results are satisfying as the tracking error of the MOVA end-effector quickly reduced to small values in simulations with three reference trajectories of different nature. Moreover, a 3D virtual reality scene is built to offer an intuitive demonstration of simulated MOVA system behavior controlled by the proposed controller in addition to plots.

Chapter 5

Conclusions

In this work, the Manipulator on VTOL Aircraft (MOVA) system, was introduced as an innovative, highly efficient mobile manipulator. The MOVA system was proposed to autonomously perform field tasks that have risks to human participation such as height or hazardous material exposure. A significant constraint in UAV design is the stringent weight budget for a system that remains airborne for the whole task period. The key aspect of the MOVA philosophy, to minimize the weight of the MOVA through combining the VTOL and manipulator degrees-of-freedom to produce a minimal kinematic design, was demonstrated. That is, the design approach uses an onboard manipulator with minimum number of joints and “borrows” degrees-of-freedom from the VTOL aircraft, which is also the base of the manipulator, to gain the ability to place the end-effector at arbitrary 3D positions at any orientation.

Preliminary investigation of a planar MOVA system dynamics and controller design was presented in preparation for developing the controller of the more complex MOVA system in 3D space. Dynamics of the planar MOVA system was derived using the Lagrangian approach and then transformed into a form that facilitates controller design using the concept of a virtual manipulator. A MOVA end-effector

trajectory tracking controller was designed with the transformed dynamics equation using integrator backstepping control design approach. Validity of the controller was shown via stability analysis, simulation results and results from a physical test-bed. The experimental results showed the potential benefit of explicit compensation of the manipulator-aircraft interaction compared to an approach with no direct compensation. The outcome of the planar MOVA system study demonstrated the feasibility of extending the research to a general MOVA system in three-dimensional space and helped define the more general technical approach.

A systematic approach was developed for the derivation of the 3D MOVA system dynamics equations. Derivation steps were presented purely with matrices and vectors to ensure conciseness of the intermediate and final results. The resulting dynamics equation is represented abstractly in the standard robotic form and is proven to have the skew-symmetric property, which is a useful property for control derivation. An open source Mathematica program, named MOVADYN, was developed to achieve automatic derivation of the MOVA system dynamics. Accessory tools were also designed to accomplish a tool-chain starting with CAD modeling, using the Mathematica system for dynamics derivation, and finishing with automatically generated Simulink diagram and MATLAB code for simulation and model-based controller implementation. The dynamics equations derived using MOVADYN were successfully validated through both analytical and numerical means so that they can be used in controller development. The derivation approach and tool-chain can support other researchers in this field.

Finally, a unified nonlinear algorithm that controls the 3D MOVA system, including both the aircraft and the onboard manipulator as single entity, was developed to achieve trajectory tracking of the MOVA end-effector position and attitude based on the explicit dynamics equation. Globally Uniformly Ultimately Bounded

(GUUB) stability is proven for the controller using Lyapunov-type stability analysis. Simulation testing was also performed in order to evaluate the performance of the proposed controller on a MOVA systems with a two-link onboard manipulator, and yields satisfying results. A 3D virtual reality scene was built to offer an intuitive demonstration of simulated MOVA system behavior.

In summation, a novel VTOL aircraft-based manipulator, named MOVA (Manipulator on VTOL Aircraft), was introduced, and research about its dynamics and control design was performed progressing from a planar case study to a general system in 3D space. Validation of the work, both analytically and using numerical tools, was provided at each step to ensure the correctness. The systematic dynamics derivation steps along with the resulting derivation program are useful for developing of MOVA system with different configurations or even systems that share similar features. The proposed end-effector trajectory tracking controller is demonstrated to offer satisfying results. This controller can be used in future physical implementations of MOVA systems or as the inner-loop of a higher-level task controller in MOVA application research.

5.1 Future Work

There is a great potential in extending the work presented in this dissertation since MOVA is still a new concept and the current results are promising. The next step will be to perform physical experiments in a test-bed of the 3D MOVA system. Innovation and optimization of the MOVA mechanical design can be done with guidance from the results of the automatic dynamics derivation program and the experimental test-bed. Extensions of the control algorithm can be performed on the basis of of the proposed work: adaptive controller may be developed to accom-

modate the presence of a payload; and a force control design will add compliance with the environment. Trajectory generation for MOVA system with redundant or multiple manipulators can be investigated for controller implementation. Teleoperation related research is another direction for innovation in which investigation of human-interface and haptics feedback for the operator take places. In order to enable in-the-field operation of MOVA system, navigation and vision sensor integration should also be performed. The long term goal for MOVA research will be adoption of MOVA systems in field applications that currently pose various types of hazards to workers and tasks that have already been done by mobile manipulators or UAV but suffer from limitations in maneuverability or ability to interact with the surrounding world.

Appendices

Appendix

Appendix A Example Usage of MOVADYN and Its Auxiliary Programs

A.1 Planar MOVA System Dynamics and Code Generation

This is an example of using MOVADYN program to derive the dynamics equation of the planar MOVA system described in Chapter 2. Basic definitions are declared in the beginning (lines 4,5) to enhance the readability of the rest of code: “u0” is zero vector; “ux”, “uy” and “uz” are unit vectors along x, y and z axis; “O3x3” is 3-by-3 zero matrix and “I3” is the 3-by-3 identity matrix. Variables Nlink, Ms, Is, ls, rs, Rs, vs are initialized according to the design of the planar MOVA from line 6 to 19. Notice that symbols, such as g, la, m0, etc., are used in some parameters and these symbols will appear in the output matrices and vectors. On line 20, the dynamics derivation is performed and the result is store in Dyn. Line 25 to 29 shows the method to retrieve separate matrix or vector from the result. The rest of the code shows steps to invoke the MOVA code generator using derived dynamics.

```
1 (* import the package to use *)
2 << MOVADynamics';
3 (*basic definition *)
```

```

4  u0 = {0, 0, 0}; ux = {1, 0, 0}; uy = {0, 1, 0}; uz = {0, 0, 1};
5  O3x3 = ConstantArray[0, {3, 3}]; I3 = IdentityMatrix[3];
6  (* Identifier of the system *)
7  MOVAType = "planar";
8  (*number of links, cannot be symbol *)
9  Nlink = 2;
10 (*masses of links*)
11 Ms = {m0, 0, ma};
12 (* moment of inertia of links *)
13 Is = { DiagonalMatrix[{I0x, I0y, I0z}], O3x3,
14        DiagonalMatrix[{Iax, Iay, Iaz}]];
15 (* r and l vectors *)
16 rs = {u0, u0, 1/2*l1*uz};
17 ls = {u0, u0, -1/2*l1*uz};
18 (* rotation axis, cannot be symbolic*)
19 vs = {uy, ux};
20 (* Initial pose*)
21 Rs = {I3, I3};
22 (* Derive dynamics and kinematics *)
23 Dyn = DeriveMovaDynamics[Nlink, Ms, Is, ls, rs, Rs, vs];
24 (* Retrieving result *)
25 H = Dyn["H"];
26 Cp = Dyn["Cp"];
27 Gb = Dyn["Gb"];
28 Jp = Dyn["Jp"];
29 FwdKin = Dyn["FwdKin"];
30 (* Write out Dynamics and Kinematics MATLAB functions *)
31 WriteMovaDynamics[MOVAType, NotebookDirectory[], NotebookDirectory[], Dyn];
32 (* Write out Parameter MATLAB functions *)
33 WriteMovaParameterFunction[MOVAType, NotebookDirectory[], Dyn];
34 (* Write out Simulink block *)

```

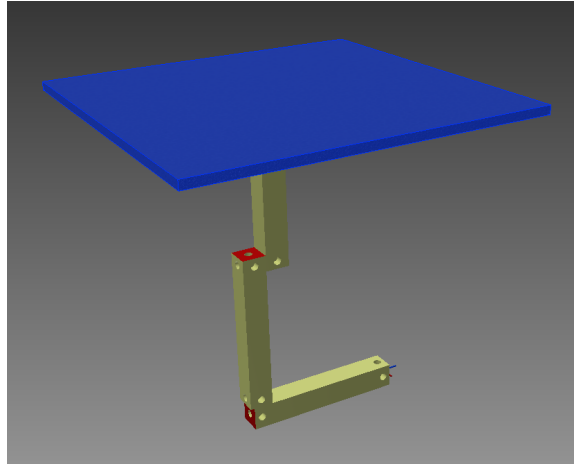


Figure A.1: 3D model of a MOVA system with a RRR onboard manipulator.

35 `WriteMovaSimulinkBlock[MOVAType, NotebookDirectory[], NotebookDirectory[]];`

A.2 Example of Parameter Exporter Usage

A hypothetical MOVA system with onboard manipulator consist of 3 links in RRR configuration is shown in Fig. A.1 as an example for illustrating usage of the physical parameter exporter. The blue object represents aircraft body and the three golden cuboids represent the three links of the manipulator. The aircraft body and three links are connected together with three “insert” constraints and relative poses of them are set by extra constraints to form the appearance in Fig. A.1. After execute the physical parameter exporter, a Mathematica script file is generated with its content shown below. All necessary parameters needed for MOVADYN derivation are populated according to the physical properties of the 3D model. Statements that invoke MOVADYN and the MOVA code generator are appended at last, enables one step code generation by executing this script in Mathematica.

```
1 MOVAType = "TEST";
2 Nlink = 3;
```

```

3  Ms = {2.74496, 0.02152, 0.02152, 0.02152, 0};
4  Is = {DiagonalMatrix[{567.03497, 6.23781, 572.12637}],
5  DiagonalMatrix[{0.17125, 0.17125, 0.00821}],
6  DiagonalMatrix[{0.17125, 0.17125, 0.00821}],
7  DiagonalMatrix[{0.17125, 0.00821, 0.17125}],
8  DiagonalMatrix[{0,0,0}]/10000;
9  ls = {{0, 0, 0},
10 {0.75000, 0, 4.50000},
11 {0, 0.75000, 4.50000},
12 {0, -4.50000, 0.75000},
13 {0,0,0}}/100;
14 rs = {{-0.24876, 0, -1.48568},
15 {0, -0.75000, -4.50000},
16 {0, 0, -5},
17 {0, 0, 0},
18 {0,0,0}}/100;
19 vs = {{-1, 0, 0},
20 {0, -1, 0},
21 {0, 0, -1},
22 {0, 0, 0},
23 {0,0,0}};
24 Rs = {
25 {1, 0, 0},
26 {0, 1, 0},
27 {0, 0, 1}},
28 {1, 0, 0},
29 {0, 1, 0},
30 {0, 0, 1}},
31 {1, 0, 0},
32 {0, 1, 0},
33 {0, 0, 1}},

```

```

34 {      {1, 0, 0},
35      {0, 1, 0},
36      {0, 0, 1}},
37 { {0,0,0},{0,0,0},{0,0,0} } };
38
39 << MOVADynamics';
40 Dyn = DeriveMovaDynamics[Nlink, Ms[[1;;Nlink+1]], Is[[1;;Nlink+1]], ls [[1;; Nlink+1]],
      rs [[1;; Nlink+1]], Rs [[1;; Nlink ]], vs [[1;; Nlink ]]];
41 WriteMovaDynamics[MOVAType, NotebookDirectory[], NotebookDirectory[], Dyn];
42 WriteMovaParameterFunction[MOVAType, NotebookDirectory[], Dyn];
43 WriteMovaSimulinkBlock[MOVAType, NotebookDirectory[], NotebookDirectory[]];

```

Bibliography

- [1] United States Dept. of Labor. (2012, Sep) Fatal occupational injuries and workers memorial day. United States Dept. of Labor, Bureau of Labor Statistics. [Online]. Available: http://www.bls.gov/iif/oshwc/cfoi/worker_memorial.htm
- [2] N. Michael and D. Mellinger, “The GRASP multiple micro-UAV testbed,” *Robotics & Automation Magazine*, vol. 17, no. 3, pp. 56 – 65, 2010.
- [3] D. Mellinger, N. Michael, and V. Kumar, “Trajectory generation and control for precise aggressive maneuvers with quadrotors,” *The International Journal of Robotics Research*, vol. 31, no. 5, pp. 664–674, Jan. 2012.
- [4] D. Lee, C. Nataraj, T. Burg, and D. Dawson, “Adaptive tracking control of an underactuated aerial vehicle,” in *Proc. IEEE American Control Conference*, 2011, pp. 2326–2331.
- [5] M. Huang, B. Xian, C. Diao, K. Yang, and Y. Feng, “Adaptive tracking control of underactuated quadrotor unmanned aerial vehicles via backstepping,” in *Proc. IEEE American Control Conference*, 2010, pp. 2076–2081.
- [6] D. Lee, C. Nataraj, and T. C. Burg, “Coordinated Control of Flying Robotic Arm,” in *Volume 3: Renewable Energy Systems; Robotics; Robust Control; Single Track Vehicle Dynamics and Control; Stochastic Models, Control and Algorithms in Robotics; Structure Dynamics and Smart Structures*;. ASME, Oct. 2012, pp. 119–127.
- [7] C. Korpela, M. Orsag, M. Pekala, and P. Oh, “Dynamic stability of a mobile manipulating unmanned aerial vehicle,” in *2013 IEEE International Conference on Robotics and Automation*. Karlsruhe, Germany: IEEE, May 2013, pp. 4922–4927.
- [8] C. Korpela, P. Brahmabhatt, M. Orsag, and P. Oh, “Towards the realization of mobile manipulating unmanned aerial vehicles (MM-UAV): Peg-in-hole insertion tasks,” in *2013 IEEE Conference on Technologies for Practical Robot Applications (TePRA)*. Woburn, MA: IEEE, Apr. 2013, pp. 1–6.

- [9] J. R. Thomas, J. J. Polin, K. Sreenath, and V. Kumar, "DRAFT: Avian-inspired Grasping for Quadrotor Micro UAVs," in *Proceedings of the ASME 2013 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*. Portland, OR: IDETC/CIE, 2013.
- [10] D. Mellinger, Q. Lindsey, M. Shomin, and V. Kumar, "Design, modeling, estimation and control for aerial grasping and manipulation," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. San Francisco, CA: Ieee, Sep. 2011, pp. 2668–2673.
- [11] P. E. I. Pounds, D. R. Bersak, and A. M. Dollar, "Grasping from the air: Hovering capture and load stability," in *2011 IEEE International Conference on Robotics and Automation*. Shanghai, China: Ieee, May 2011, pp. 2491–2498.
- [12] H. Goldstein, C. P. Poole, and J. L. Safko, "Variational Principles and Lagrange's Equations," in *Classical Dynamics*, 3rd ed. Addison-Wesley, 2001, ch. 2, pp. 45–51.
- [13] D. T. Greenwood, "Hamilton's Equation," in *Classical Dynamics*. Courier Dover Publications, 1997, pp. 147–164.
- [14] M. W. Spong, S. Huthinson, and M. Vidyasagar, "Dynamics," in *Robot Modeling and Control*. Springer, 2005, ch. 9.
- [15] E. Papadopoulos, "On the dynamics and control of space manipulators," Ph.D. dissertation, Massachusetts Institute of Technology, 1990.
- [16] Z. Vafa and S. Dubowsky, "On the dynamics of manipulators in space using the virtual manipulator approach," in *Proc. IEEE Int. Conf. on Robotics and Automation*, vol. 4. Institute of Electrical and Electronics Engineers, 1987, pp. 579–585.
- [17] S. A. A. Moosavian and E. Papadopoulos, "Explicit dynamics of space free-flyers with multiple manipulators via SPACEMAPLE," *Advanced Robotics*, vol. 18, no. 2, pp. 223–244, Jan. 2004.
- [18] I. Schjø lberg and T. I. Fossen, "Modelling and control of underwater vehicle-manipulator systems," *Conf. on Marine Craft Maneuvering and Control*, pp. 45–57, 1990.
- [19] G. Antonelli, F. Caccavale, S. Chiaverini, and L. Villani, "Tracking control for underwater vehicle-manipulator systems with velocity estimation," *IEEE Journal of Oceanic Engineering*, vol. 25, no. 3, pp. 399–413, Jul. 2000.
- [20] K. Leabourne and S. Rock, "Model development of an underwater manipulator for coordinated arm-vehicle control," *Proc. IEEE Oceanic Engineering Society OCEANS'98*, vol. 2, pp. 941–946, 1998.

- [21] G. Wood and D. Kennedy, "Simulating mechanical systems in Simulink with SimMechanics," The Mathworks, Tech. Rep., 2003.
- [22] S. Dubowsky, "Path planning for space manipulators to minimize spacecraft attitude disturbances," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, vol. 3, Sacramento, 1991, pp. 2522–2528.
- [23] K. Yoshida, "Engineering Test Satellite VII Flight Experiments for Space Robot Dynamics and Control: Theories on Laboratory Test Beds Ten Years Ago, Now in Orbit," *The Int. Journal of Robotics Research*, vol. 22, no. 5, pp. 321–335, May 2003.
- [24] S. Dubowsky and E. Papadopoulos, "The kinematics, dynamics, and control of free-flying and free-floating space robotic systems," *IEEE Trans. Robot. Autom.*, vol. 9, no. 5, pp. 531–543, 1993.
- [25] B. Erginer and E. Altug, "Modeling and PD Control of a Quadrotor VTOL Vehicle," in *2007 IEEE Intelligent Vehicles Symposium*. IEEE, Jun. 2007, pp. 894–899.
- [26] S. Bouabdallah, A. Noth, and R. Siegwart, "PID vs LQ control techniques applied to an indoor micro quadrotor," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, vol. 3. IEEE, 2004, pp. 2451–2456.
- [27] P. Setlur, D. Dawson, Y. Fang, and B. Costic, "Nonlinear tracking control of the VTOL aircraft," in *Proc. of the 40th IEEE Conference on Decision and Control*, vol. 5. IEEE, 2001, pp. 4592–4597.
- [28] M. Saeki and Y. Sakaue, "Flight control design for a nonlinear non-minimum phase VTOL aircraft via two-step linearization," in *Proc. of the 40th IEEE Conference on Decision and Control*, vol. 1. IEEE, 2001, pp. 217–222.
- [29] S. Bouabdallah and R. Siegwart, "Backstepping and Sliding-mode Techniques Applied to an Indoor Micro Quadrotor," in *Proc. IEEE Int. Conf. on Robotics and Automation*. IEEE, 2005, pp. 2247–2252.
- [30] N. Guenard, T. Hamel, and R. Mahony, "A Practical Visual Servo Control for an Unmanned Aerial Vehicle," *IEEE Trans. Robotics*, vol. 24, no. 2, pp. 331–340, Apr. 2008.
- [31] E. Altug, J. Ostrowski, and R. Mahony, "Control of a quadrotor helicopter using visual feedback," in *Proc. IEEE Int. Conf. on Robotics and Automation*, vol. 1. IEEE, 2002, pp. 72–77.
- [32] S. Joshi, A. Kelkar, and J.-Y. Wen, "Robust attitude stabilization of spacecraft using nonlinear quaternion feedback," *IEEE Transactions on Automatic Control*, vol. 40, no. 10, pp. 1800–1803, 1995.

- [33] D. Fragopoulos and M. Innocenti, “Stability considerations in quaternion attitude control using discontinuous Lyapunov functions,” *IEE Proceedings-Control Theory and Applications*, pp. 253–258, 2004.
- [34] R. Kristiansen and P. Nicklasson, “Satellite attitude control by quaternion-based backstepping,” in *Proceedings of the 2005, American Control Conference, 2005*. IEEE, 2005, pp. 907–912.
- [35] R. Kristiansen, P. Nicklasson, and J. Gravdahl, “Satellite Attitude Control by Quaternion-Based Backstepping,” *IEEE Transactions on Control Systems Technology*, vol. 17, no. 1, pp. 227–232, Jan. 2009.
- [36] V. Chitrakaran, D. Dawson, and M. Feemster, “Vision Assisted Autonomous Landing of an Unmanned Aerial Vehicle.” in *Proc. of the 44th IEEE Conf. on Decision and Control*. IEEE, 2005, pp. 1465–1470.
- [37] R. Huang, “Design and demonstration of a two-dimensional testbed for UAV controller evaluation,” Ph.D. dissertation, Clemson University, 2014.
- [38] J. Wittenburg, “General Multibody Systems,” in *Dynamics of Multibody Systems*, 2nd ed. New York,: Springer, 2008, pp. 89–188.
- [39] R. Seifried, P. Eberhard, and T. Kurz, “Simulation and Control Design of Flexible Multibody Systems Using Neweul-M2,” Institute of Engineering and Computational Mechanics, University of Stuttgart, Stuttgart, Germany, Tech. Rep. 1, 2010.
- [40] T. Kurz, “Neweul-M2 Symbolic multibody simulation in Matlab,” 2011. [Online]. Available: <http://www.itm.uni-stuttgart.de/research/neweul/media/Neweulm2.Manual.pdf>
- [41] S. Ploen, “A skew-symmetric form of the recursive Newton-Euler algorithm for the control of multibody systems,” in *Proceedings of American Control Conference , 1999*, no. June. IEEE, 1999, pp. 3770–3773.
- [42] P. Xu, “MOVADYN Opensource Project.” [Online]. Available: <https://bitbucket.org/sippey/movadyn>
- [43] “Creative Commons Attribution 4.0 International.” [Online]. Available: <http://creativecommons.org/licenses/by/4.0/>
- [44] H. Ojanen, “No Title,” 1999. [Online]. Available: <http://library.wolfram.com/infocenter/MathSource/577/>
- [45] R. W. Brockett, “Asymptotic Stability and Feedback Stabilization,” Harvard University, Tech. Rep., 1983.

- [46] P. Xu, R. Huang, D. Lee, and T. C. Burg, “Dynamics and Control of a Novel Manipulator on VTOL Aircraft (MOVA) System - A Planar Case Study,” in *Proc. IEEE American Control Conference*, Portland, OR, 2014.
- [47] C. G. Mayhew, R. G. Sanfelice, and A. R. Teel, “Quaternion-Based Hybrid Control for Robust Global Attitude Tracking,” *IEEE Transactions on Automatic Control*, vol. 56, no. 11, pp. 2555–2566, Nov. 2011.