8-2009

LYAPUNOV-BASED COORDINATED CONTROL OF AN UNDERACTUATED UNMANNED AERIAL VEHICLE AND ROBOT MANIPULATOR

DongBin Lee
Clemson University, edongbean@gmail.com

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

Recommended Citation
https://tigerprints.clemson.edu/all_dissertations/502

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.
LYAPUNOV-BASED COORDINATED CONTROL
OF AN UNDERACTUATED UNMANNED AERIAL
VEHICLE AND ROBOT MANIPULATOR

A Dissertation
Presented to
the Graduate School of
Clemson University

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

By
DongBin Lee
August 2009

Accepted by:
Dr. Timothy C. Burg, Committee Chair
Dr. Darren M. Dawson, Co-Advisor
Dr. Ian D. Walker,
Dr. John R. Wagner
ABSTRACT

This Ph.D. dissertation describes nonlinear tracking control results for a quadrotor helicopter unmanned aerial vehicle (UAV) towards the ultimate goal of controlling a combined UAV plus robot manipulator system (UAVRM). The quadrotor UAV is a helicopter that has four independent rotors that provide vertical lift: these four independent forces are managed in order to directly provide lift, pitch, roll, and yaw of the vehicle. Horizontal translations result from pitch and roll actions, the system is underactuated in the sense that there are only four control inputs to move the six degree-of-freedom aircraft. There are existing dynamic models of the quadrotor UAV that relate the input forces and torques to position, velocity, and acceleration. Use of these dynamic models for model-based UAV control design is explored. First, a parametric uncertain model of the UAV system was considered. A robust control approach is proposed to account for the fact that the model parameters are difficult to measure exactly in a physical system. The controller uses full state feedback signals and a robust control scheme is designed to compensate for the unknown parameters in each dynamic subsystem model using a Lyapunov-based approach. Lyapunov-type stability analysis suggests a global uniform ultimately bounded (GUUB) tracking result. Next, the difficulties of UAV state measurement is considered; specifically, where only the output position signals are available but no velocities or acceleration signals are measurable. The output feedback control proposes a new control approach for trajectory-tracking by the quadrotor family of small-scale unmanned aerial vehicles (UAV), in which only the positions and yaw angle are measured. The tracking control result is achieved using an observer, which estimates velocity signal based on exact knowledge of the dynamic modeling of equation. An integrator backstepping approach is applied to this cascaded and coupled nonlinear dynamic system to perform an observer and closed-loop controller design via a Lyapunov-type analysis. A semi-global, uniformly ultimate bounded (SGUUB) tracking result is achieved.
The application of remote robots equipped with a robotic hands or arms, has been growing in applications where it is dangerous or inconvenient to use direct human intervention. Recently, the area of unmanned aerial robot system has seen an amazing growth in both military and civilian applications. UAVs have the distinct advantage of being able to move rapidly, free of ground obstacles. Most current applications of the aerial robot system use the UAV as “eye-in-the-sky” for surveillance and monitoring applications. Projecting the intersection of these two trends, if the integration of robot manipulator and aerial robot system is possible, the system could be fast moving and avoid obstacles, but also useful for manipulating physical systems once in place, e.g., changing a lightbulb on a radio tower. This is the ultimate purpose of the work contained in this dissertation - the development of the unmanned aerial-robotic system. A model of the combined UAV and robot manipulator is proposed from which a coordinated controller of the integrated nonlinear system is developed using a Lyapunov-type method. The design goal for this controller is to simultaneously control the two degree-of-freedom robot manipulator (RM) and the quadrotor Unmanned Aerial Vehicle (UAV) to create a six degree-of-freedom UAV-Robot Manipulator (UAVRM). The UAVRM end-effector can track three desired positions and three angles using feedback signals.
ACKNOWLEDGEMENTS

I would like to acknowledge those who helped this dissertation. I would like to deeply express my thanks to Dr. Timothy C. Burg for his mentoring and help several years in the UAV and Haptics Labs where I had enjoyed my research. I would like to also express my gratitude to Dr. Darren M. Dawson for his inspiring guidance, confidence, and support during my Ph.D work in the Control and Robotics Lab. I am thankful for all the opportunities I had been given to contribute to a variety of projects and research of the research groups. I am grateful to Dr. Ian D. Walker for his lectures and Dr. John R. Wagner for his advice as my committee. I am happy to convey my thanks to all the members I have been working at the groups especially, Dr. Bin Xian and Mr. Shu. I would like to appreciate Dr. Hwa Young Yim’s help to open my eyes to the learning and thanks to Dr. Nataraj for his encouragement for future works. Finally, I couldn’t have done this work without the help of my family; my wife, my son, my spiritual mom, Pat Wannamaker, and the Bible study members with His easy yoke.
DEDICATION

I dedicate this to my wife, Carol (HyeRyeon), my son, Daniel (Junhyeong), and God.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>TITLE PAGE</td>
<td>i</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>ii</td>
</tr>
<tr>
<td>ACKNOWLEDGEMENTS</td>
<td>iv</td>
</tr>
<tr>
<td>DEDICATION</td>
<td>v</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>viii</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>ix</td>
</tr>
<tr>
<td>CHAPTER</td>
<td></td>
</tr>
<tr>
<td>1. OVERVIEW</td>
<td>1</td>
</tr>
<tr>
<td>2. ROBUST TRACKING CONTROL</td>
<td>3</td>
</tr>
<tr>
<td>Introduction</td>
<td>3</td>
</tr>
<tr>
<td>System Model</td>
<td>4</td>
</tr>
<tr>
<td>Tracking Controller Design</td>
<td>5</td>
</tr>
<tr>
<td>Lyapunov-based Robust Control</td>
<td>10</td>
</tr>
<tr>
<td>Simulation and Implementation</td>
<td>16</td>
</tr>
<tr>
<td>Conclusion</td>
<td>18</td>
</tr>
<tr>
<td>BIBLIOGRAPHY</td>
<td>22</td>
</tr>
<tr>
<td>3. OUTPUT FEEDBACK TRACKING CONTROL</td>
<td>23</td>
</tr>
<tr>
<td>Introduction</td>
<td>23</td>
</tr>
<tr>
<td>System Model</td>
<td>25</td>
</tr>
<tr>
<td>Output Feedback Tracking Controller Design</td>
<td>28</td>
</tr>
<tr>
<td>Stability Analysis</td>
<td>36</td>
</tr>
<tr>
<td>Simulation</td>
<td>36</td>
</tr>
<tr>
<td>Conclusion</td>
<td>40</td>
</tr>
<tr>
<td>BIBLIOGRAPHY</td>
<td>42</td>
</tr>
</tbody>
</table>
Table of Contents (Continued)

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4. COORDINATED CONTROL</td>
<td>44</td>
</tr>
<tr>
<td>UAVRM Kinematics</td>
<td>44</td>
</tr>
<tr>
<td>UAVRM End-Effector Kinematics</td>
<td>49</td>
</tr>
<tr>
<td>Integrated System Modeling</td>
<td>52</td>
</tr>
<tr>
<td>Coordinated Tracking Control</td>
<td>56</td>
</tr>
<tr>
<td>Conclusion</td>
<td>61</td>
</tr>
<tr>
<td>BIBLIOGRAPHY</td>
<td>63</td>
</tr>
<tr>
<td>Future Work</td>
<td>64</td>
</tr>
<tr>
<td>Conclusion</td>
<td>64</td>
</tr>
<tr>
<td>Notation and Nomenclature</td>
<td>64</td>
</tr>
<tr>
<td>APPENDICES</td>
<td>66</td>
</tr>
<tr>
<td>A. Rigid-body Equation of Motion</td>
<td>67</td>
</tr>
<tr>
<td>Quadrotor Helicopter Translational Dynamics</td>
<td>70</td>
</tr>
<tr>
<td>B. Equations of Motion of Two-Link Robot Manipulator</td>
<td>77</td>
</tr>
<tr>
<td>C. Rotations of the UAVRM</td>
<td>91</td>
</tr>
<tr>
<td>1. Euler Angles (Roll, Pitch, and Yaw) Rotation Matrix</td>
<td>91</td>
</tr>
<tr>
<td>2. UAVRM Total System Coordinate Frames and Rotations</td>
<td>95</td>
</tr>
<tr>
<td>BIBLIOGRAPHY</td>
<td>100</td>
</tr>
</tbody>
</table>
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1. Denavit-Hartenberg Convention for UAV and Robot Manipulator</td>
<td>44</td>
</tr>
<tr>
<td>B.1. Denavit-Hartenberg Table for 2-link Robot Manipulator Kinematics</td>
<td>80</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Robust Tracking Demonstration</td>
<td>19</td>
</tr>
<tr>
<td>2.2</td>
<td>Position ((x, y, z)) and Yaw Angle ((\psi)) Tracking at Each Axis</td>
<td>19</td>
</tr>
<tr>
<td>2.3</td>
<td>Tracking Errors in Position ((x, y, z)) and Yaw Angle ((\psi)) at Each Axis</td>
<td>20</td>
</tr>
<tr>
<td>2.4</td>
<td>Force about Z-axis and Torques about Roll, Pitch, and Yaw</td>
<td>20</td>
</tr>
<tr>
<td>3.1</td>
<td>Output Feedback Trajectory-Tracking Demonstration. The command and result are split into two plots to show the rise of the UAV (left) and the descent (right)</td>
<td>37</td>
</tr>
<tr>
<td>3.2</td>
<td>3D Position and Yaw Angle Tracking (left) and Error (right) at Each Axis</td>
<td>37</td>
</tr>
<tr>
<td>3.3</td>
<td>Control inputs; the translational force input (u_1(t)) and the torque commands (u_2(t))</td>
<td>38</td>
</tr>
<tr>
<td>3.4</td>
<td>Estimated Velocity Output of the Velocity Observer</td>
<td>38</td>
</tr>
<tr>
<td>3.5</td>
<td>Gyroscopic Forces of the Rolling, Pitching, and Yawing</td>
<td>39</td>
</tr>
<tr>
<td>4.1</td>
<td>Integrated UAV and Robot Manipulator (UAVRM) Coordinate Frames</td>
<td>45</td>
</tr>
<tr>
<td>A.1</td>
<td>Rigid-body Coordinate Frames: Earth-Fixed Inertial, Body-fixed, and Center of Gravity</td>
<td>68</td>
</tr>
<tr>
<td>A.2</td>
<td>Quadrotor Helicopter Coordinate Frames</td>
<td>71</td>
</tr>
<tr>
<td>B.1</td>
<td>2-link Robot Manipulator</td>
<td>78</td>
</tr>
<tr>
<td>C.1</td>
<td>Rotation Matrix about X-axis by an Angle of (\phi)</td>
<td>91</td>
</tr>
<tr>
<td>C.2</td>
<td>Rotation ((\theta)) Matrix about Y-axis</td>
<td>92</td>
</tr>
<tr>
<td>C.3</td>
<td>Rotation ((\psi)) Matrix about Z-axis</td>
<td>94</td>
</tr>
<tr>
<td>C.4</td>
<td>UAVRM Integrated System Configuration</td>
<td>96</td>
</tr>
</tbody>
</table>
List of Figures (Continued)

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>C.5</td>
<td></td>
<td>96</td>
</tr>
<tr>
<td>C.6</td>
<td>Homogeneous Transformation Matrix: $A_1$ for the First Link</td>
<td>97</td>
</tr>
<tr>
<td>C.7</td>
<td>Homogeneous Transformation Matrix: $A_2$ for the Second Link</td>
<td>98</td>
</tr>
<tr>
<td>C.8</td>
<td>Homogeneous Transformation Matrix: $A_3$ for the Third Link</td>
<td>99</td>
</tr>
</tbody>
</table>
CHAPTER 1
OVERVIEW

This thesis is divided into three main chapters and two appendices. Chapter 2 covers the control of an underactuated quadrotor to obtain position tracking along the x-, y-, and z-axes and the yaw angle in the presence of parametric uncertainty. The definitions of error signals are developed based on state feedback tracking control, a backstepping approach is introduced to the dynamic model of the quadrotor, and a robust controller for compensating the unknown constant parameters is suggested.

In Chapter 3, a tracking controller for desired 3D position and yaw commands is designed using the nonlinear dynamic model of the quadrotor helicopter and only output feedback signals. The yaw angle controller will provide an extra degree of freedom for tasks such as surveillance or object targeting. An observer, velocity estimator, is designed to compensate for the unmeasurable signals. The observer and controller are designed concurrently via Lyapunov stability arguments in order to ensure a stability result. The choice here to control the three linear translations requires that some of the desired translational commands be “reformulated” as attitude commands via the backstepping approach. An integrator backstepping method in which the virtual force command is backstepped through the integrator to the rotational controller is introduced in order to achieve the tracking control objectives. The novelty of this work is that we propose a trajectory tracking control approach that uses only position and angle measurements and is fully supported by a mathematical stability proof. The payoff from this approach is the mathematical assertion of semi-global, uniformly ultimately bounded tracking while obtaining the control objectives.

Chapter 4 details the design of a controller for the nonlinear coordinated unmanned system, UAV, with an attached robot manipulator using Lyapunov-type stability methods. After modeling of each system, we integrate both mechanisms into a single dynamic, interconnected model. The design goal for this controller is to add a two degree-of-freedom robot manipulator (RM) to the quadrotor Unmanned Aerial
Vehicle (UAV) to create a single UAV-Robot Manipulator (UAVRM). The UAVRM end-effector can track three desired position and three angles using feedback signals.

The integrated control of the UAV and robot manipulator requires modeling equation for each subsystem. Appendix A includes a brief overview of modeling of the quadrotor rigid-body and Appendix B describes the equations of motion for a two-link robotic arm in support of the model development in Chapter 4. The quadrotor UAV model is combined with robot manipulator equation of motion to create the integrated model of the UAVRM in Chapter 4.
CHAPTER 2
ROBUST TRACKING CONTROL

Introduction

The quadrotor helicopter system is underactuated in the sense that the system is considered to have four control inputs to move and position the six degree-of-freedom system. Although the system has four independently controllable motor-rotor sets, a built-in strategy to group these motor-rotor sets to create vertical thrust, and roll, pitch, and yaw torques is presumed. An important property of the quadrotor system is that it can be modeled as coupled translational and angular rotational subdynamics where the output of the rotational dynamics are cascaded as the input to the translational dynamics. The coupled and cascaded dynamic model is used as the basis for the control design. The coupling between linear and angular dynamic subsystems occurs in the gyroscopic terms which are typically neglected in a hovering model.

Many researchers have proposed a variety control solutions for the underactuated quadrotor system. Of special significance, in [13], the control compensates for wind affects acting on the underactuated quadrotor. The work in [3] presented model-based control techniques applied to an underactuated quadrotor. Of particular note, the system dynamics include nonlinearities in the aerodynamic forces. The authors in [1] proposed a solution of the trajectory-tracking and path-following for underactuated autonomous vehicles such as underwater vehicle and a hovercraft in the presence of parametric modeling uncertainty.

A feasible control approach when the dynamic model includes gyroscopic effects is to use a backstepping approach. The work in [7] presented a trajectory tracking controller for an underactuated small helicopter using a backstepping procedure. In [8], a backstepping approach to control a specific model of a quadrotor, the X4 flyer, is
presented. This work includes the dynamic complication of the aerodynamic and gyroscopic effects of the rotating blades. The work in [4] presented attitude stabilization of the quadrotor aircraft using the backstepping technique.

This research emphasizes the control of an underactuated quadrotor to obtain position tracking along the x-, y-, and z-axes and the yaw angle in the presence of parametric uncertainty. In Section 2, a brief outline of a quadrotor-helicopter model is presented along with a property and assumptions used in the control design. In Section 3, the definitions of error signals are developed based on state feedback tracking control, a backstepping approach is introduced to the dynamic model of quadrotor, and a robust controller for compensating the unknown constant parameters is suggested. Stability analysis on the controller is considered in Section 4 followed by a theorem. Concluding remarks are presented in Section 6.

System Model

The translational and rotational kinematic equations of the quadrotor unmanned helicopter are given [12] by

\[
\begin{bmatrix}
v \\
\omega
\end{bmatrix} = \begin{bmatrix}
R^T(\Theta) & O_{3\times3} \\
0_{3\times3} & T^{-1}(\Theta)
\end{bmatrix} \begin{bmatrix}
\dot{p} \\
\dot{\Theta}
\end{bmatrix} \in \mathbb{R}^6
\] (2.1)

where \(v(t), \omega(t) \in \mathbb{R}^3\) denote the linear velocity and the angular velocity, respectively. The Euler-based rotation matrix \(R(\Theta) = R_{\phi,t} \cdot R_{\theta,t} \cdot R_{\psi,t} \in SO(3)\) that translates a body-fixed frame referenced quantity into Earth-fixed inertial coordinates for yaw, pitch, and roll in order (refer to [9]) is calculated from

\[
R(\Theta) = \begin{bmatrix}
c\psi c\theta & c\psi s\theta s\phi - s\psi c\phi & s\psi s\phi + c\psi c\phi s\theta \\
s\psi c\theta & c\psi c\phi + s\phi s\theta s\psi & s\theta s\phi c\psi - c\psi s\phi \\
-s\theta & c\theta s\phi & c\theta c\phi
\end{bmatrix}
\] (2.2)

where \(\Theta(t) = [\phi, \theta, \psi]^T \in \mathbb{R}^3\) are the Euler angles, \(c \cdot = \cos(\cdot)\) and \(s \cdot = \sin(\cdot)\) are used. The body-fixed angular velocities are transformed by the matrix \(T(\Theta) \in \mathbb{R}^{3 \times 3}\), into the inertial frame [3] and is given by

\[
T(\Theta) = \begin{bmatrix}
T_x(\Theta) \\
T_y(\Theta) \\
T_z(\Theta)
\end{bmatrix} = \begin{bmatrix}
1 & s\phi t\theta & c\phi t\theta \\
0 & c\phi & -s\phi \\
0 & s\phi/c\theta & c\phi/c\theta
\end{bmatrix}.
\] (2.3)
The dynamic model of the underactuated quadrotor is given in the body-fixed reference frame by

\[
\begin{bmatrix}
    mI_3 & O_{3 \times 3} \\
    O_{3 \times 3} & J
\end{bmatrix}
\begin{bmatrix}
    \dot{v} \\
    \dot{\omega}
\end{bmatrix}
= \begin{bmatrix}
    -mS(\omega) & O_{3 \times 3} \\
    O_{3 \times 3} & -S(\omega)J
\end{bmatrix}
\begin{bmatrix}
    v \\
    \omega
\end{bmatrix}
- \begin{bmatrix}
    N_1(\theta_1, v, |v|) \\
    N_2(\theta_2, v, |v|)
\end{bmatrix}
+ \begin{bmatrix}
    G(R) \\
    B_1, O_{3 \times 3} \\
    O_{3 \times 1}, B_2
\end{bmatrix}
\begin{bmatrix}
    u_1 \\
    u_2
\end{bmatrix}
\tag{2.4}
\]

where \( N_1(\theta_1, v, |v|) \) are aerodynamic forces on the rigid-body and \( N_2(\theta_2, v, |v|) \) denotes aerodynamic induced moments where \( \theta_i \) is a parameter vector and \(|v|\) is the norm of linear velocity vector \( v(t) \), and \(-S(\omega)Jw = S(J\omega)w\). (See [12] for details). \( m \in \mathbb{R}^1 \) is the mass of the quad-rotor, \( J \in \mathbb{R}^{3 \times 3} \) denotes a positive definite diagonal inertia matrix, and the gravity force is given by \( G(R) = mgR^T(\Theta)e_3 \in \mathbb{R}^3 \) where \( e_3 = [0, 0, 1]^T \) and \( g \in \mathbb{R}^1 \) denotes the gravitational acceleration due to the gravity. These are all assumed to be unknown constants including the coefficients of the aerodynamic terms. The dynamic system in (3.1) has the following property.

**P1:** The unknown system parameters are upper and lower bounded to satisfy the following inequalities

\[
\underline{\theta}_{ij} \leq \theta_{ij} \leq \overline{\theta}_{ij}
\tag{2.5}
\]

where \( \theta_i \) is the \( i \)th parameter of the \( j \)th vector \( \theta_j \). The aerodynamic forces and moments in (3.1) can be linearly parameterized in the form

\[
\begin{align*}
    N_1(\theta_1, v, |v|) & \equiv Y_1(v, |v|)\theta_1, \\
    N_2(\theta_2, v, |v|) & \equiv Y_2(v, |v|)\theta_2
\end{align*}
\tag{2.6}
\]

where \( Y_1(v, |v|) \in \mathbb{R}^{3 \times 3} \) and \( Y_2(v, |v|) \in \mathbb{R}^{3 \times 3} \) are known regression matrices, \( \theta_1 \in \mathbb{R}^3 \) and \( \theta_2 \in \mathbb{R}^3 \) are unknown constant parameter vectors.

The following assumption is made regarding the dynamic model.

**A1:** The pitch angle \((\theta)\) is not close to \( \pm \frac{\pi}{2} \) so that \( T^{-1}(\Theta) \) is invertible.
Tracking Controller Design

The goal of the tracking controller is to force the aerial vehicle to track a desired trajectory. The error formulation for the tracking is developed and a backstepping approach for the coupled and cascaded system is utilized to facilitate the controller specification. The desired trajectories and up to their third derivatives are all bounded; i.e., $p_d(t)$, $\dot{p}_d(t)$, $\ddot{p}_d(t)$, and $\dddot{p}_d(t) \in L_\infty$ and $\psi_d(t)$, $\dot{\psi}_d(t)$, and $\ddot{\psi}_d(t) \in L_\infty$.

Error System Development

The position tracking error, denoted as $e_p(t)$, is defined in the body-fixed frame as the transformed difference between the inertial-frame based position, $p(t)$, and the inertial-frame based desired position, denoted as $p_d(t) \in \mathbb{R}^3$, in the manner

$$e_p = R^T(p - p_d) \in \mathbb{R}^3. \tag{2.7}$$

The position tracking error rate, $\dot{e}_p(t) \in \mathbb{R}^3$, is obtained by taking the time derivative of (2.7), and utilizing

$$\dot{e}_p = -S(\omega)e_p + v - R^T \dot{p}_d, \tag{2.8}$$

the definition of $e_p(t)$ in (2.7), $v(t) = R^T \dot{p}$ from (3.2), and $\dot{R}^T = -S(\omega)R^T$. Note that the last two terms in (4.44) constitute the velocity error. For subsequent control development, adding and subtracting $\frac{1}{m}R^T \dot{p}_d(t)$ yields

$$\dot{e}_p = -S(\omega)e_p + \frac{1}{m}e_v + \frac{1}{m}R^T \dot{p}_d - R^T \dot{p}_d. \tag{2.9}$$

The virtual translational velocity tracking error, denoted by $e_v(t) \in \mathbb{R}^3$, in (4.49) is defined as

$$e_v = mv - R^T \dot{p}_d. \tag{2.10}$$

Note that this is not real velocity error but was manufactured for the control development. The final form of the position tracking error is obtained from (4.49) and (2.10) as follows

$$\dot{e}_p = -S(\omega)e_p + \frac{1}{m}e_v + \left(\frac{1}{m} - 1\right)R^T \dot{p}_d. \tag{2.11}$$
After taking the time derivative of \( e_v(t) \) in (2.10), substituting for \( m\ddot{v}(t) \) from (3.1), \(-S(\omega) R^{T}\) from \( \dot{R}^{T} \), and then applying the definition of \( e_v(t) \) in (2.10), we get the velocity error rate as

\[
\dot{e}_v = -S(\omega)e_v + G(R) - Y_1(v, |v|)\theta_1 - R^T \ddot{p}_d + B_1 u_1 \tag{2.12}
\]

where Property P1 was used to replace \( N_1(\theta_1, v, |v|) \). The yaw angle tracking error, \( e_{\psi}(t) \in \mathbb{R}^1 \), is defined as

\[
e_{\psi} = \psi - \psi_d. \tag{2.13}
\]

The goal in the control development will be to ensure that \( e_{\psi}(t) \) and \( e_p(t) \) are driven to small values. The yaw angle rate error is derived by taking the time derivative of (2.13) as follows

\[
\dot{e}_{\psi} = \dot{\psi} - \dot{\psi}_d = T_z(\Theta)\omega - \dot{\psi}_d \in \mathbb{R}^1 \tag{2.14}
\]

where \( T_z(\Theta) \in \mathbb{R}^{1 \times 3} \) is the third row vector of \( T(\Theta) \) from (2.3). Note that \( T_z(\Theta)\omega(t) = \dot{\psi}(t) \) in \( \dot{\Theta}(t) \) where \( \dot{\psi}_d(t) \) is the desired yaw angular velocity in the body-fixed frame.

In order to further develop the control design, the filtered position tracking error signal \( r_p(t) \in \mathbb{R}^3 \) is defined in the following manner [5]

\[
r_p = e_v + \alpha e_p + \delta \tag{2.15}
\]

where \( \alpha \in \mathbb{R}^1 \) is a positive constant and \( \delta = [0, 0, \delta_3]^T \in \mathbb{R}^3 \) is a constant design vector in which \( \delta_3 \in \mathbb{R}^1 \) is a constant. The filtered position tracking error can be combined with the yaw tracking error to create a composite tracking error \( r(t) \in \mathbb{R}^4 \) in the manner

\[
r = [r_p^T, e_{\psi}]^T. \tag{2.16}
\]

The filtered tracking error dynamics can be found by first differentiating (2.16) to yield

\[
\dot{r} = [\dot{r}_p^T, \dot{e}_{\psi}]^T = [\dot{e}_v^T + \alpha \dot{e}_p^T, \dot{e}_{\psi}]^T \in \mathbb{R}^4. \tag{2.17}
\]

The filtered position tracking error rate, \( \dot{r}_p(t) \), is obtained by substituting (3.13) and (3.14), and the term \( S(\omega)\delta \) has been added and subtracted to facilitate introduction
of \( \dot{r}_p(t) \in \mathbb{R}^3 \) on the right-hand side as

\[
\dot{r}_p = \alpha v - S(\omega)r_p - \alpha R^T \dot{p}_d - R^T \ddot{p}_d - \frac{e_p}{m} + [G(R) - Y_1 \theta_1 + \frac{e_p}{m}] + [S(\omega)\delta + B_1 u_1]
\]

(2.18)

where \( \frac{e_p}{m} \) is subtracted and added for the subsequent stability analysis and \( \alpha v - \alpha R^T \dot{p}_d = \frac{\alpha}{m} e_v + (\frac{\alpha}{m} - \alpha) R^T \ddot{p}_d \) was used for separating the measurable and unknown terms. It is now a straightforward matter to substitute from (3.15) and (3.16) into (2.17) to yield the open-loop filtered tracking error dynamics in the following form

\[
\dot{r} = \left[ \alpha v - S(\omega)r_p - \alpha R^T \dot{p}_d - R^T \ddot{p}_d + W_1 \Theta_1 \right] - \frac{1}{m} e_p + \begin{bmatrix} -\psi_d \\ -S(\delta) \\ T_z(\Theta) \end{bmatrix} \begin{bmatrix} \omega \\ B_1 \\ 0 \end{bmatrix}
\]

(2.19)

where \( S(\omega)\delta = -S(\delta)\omega \) was used [3]. Note that \( \delta \) is a bounding constant which is utilized to incorporate the coupling between translational and rotational dynamics via the matrix \( S(\delta)\omega \) in the ensuing backstepping approach, \( \dot{r}(t) \) is derived from the position error rate \( \dot{e}_p(t) \), yaw angle error rate \( \dot{e}_\psi(t) \), and the translational dynamics \( \dot{m} \dot{v}(t) \) from (3.14) where \( v(t) \) is coupled with the angular velocity \( \omega(t) \).

**P2:** The combined term \( W_1 \Theta_1 \in \mathbb{R}^3 \) in (2.19) is defined as

\[
W_1 \Theta_1 = G(R) - Y_1(v, |v|)\theta_1 + \frac{e_p}{m}
\]

(2.20)

and satisfies a linear parameterization where \( W_1 \in \mathbb{R}^{3 \times p} \) is a regression matrix, \( p \) is the number of uncertain parameters, and \( \Theta_1 \in \mathbb{R}^p \) is a constant parameter vector.

Integrator Backstepping

Note that the angular velocity in the first equation of (3.1) is obtained by integrating the angular dynamics \( \dot{\omega}(t) \) in the second equation of (3.1). Thus, this system can be seen as a cascaded and coupled system and here we suggest a backstepping approach, the reader is referred to [5] where the control of cascaded dynamic is addressed. As shown in the last term in the error dynamics in (2.19), \( \omega(t) \) and \( u_1(t) \)
should be controlled simultaneously. Equation (2.19) can be described by using a general dynamic form as

\[ \dot{r} = f_1(r) + g_1 \mu \]  

(2.21)

where the first bracketed term defines \( f_1(r) \in \mathbb{R}^4 \), the last matrix and vector define \( g_1 \in \mathbb{R}^{4 \times 4} \), and \( \mu = [\omega^T \ u_1]^T \in \mathbb{R}^4 \). Modifying \( \mu(t) \) by a change in variables is achieved by adding and subtracting a new control signal \( g_1 \bar{u}_1(t) \) into (2.21) [4] and yields

\[ \dot{r} = f_1(r) + g_1 \bar{u}_1 + g_1(\mu - \bar{u}_1). \]  

(2.22)

Then, manipulating the last parenthetical term in (2.22) yields

\[ \mu - \bar{u}_1 = \begin{bmatrix} \omega - B_z \bar{u}_1 \\ u_1 - B_o \bar{u}_1 \end{bmatrix}. \]  

(2.23)

The actual translational control input \( u_1(t) \) is designed by

\[ u_1 = B_o \bar{u}_1 \text{ and } B_o = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \in \mathbb{R}^{1 \times 4}. \]  

(2.24)

Finally, we get

\[ \mu - \bar{u}_1 = \begin{bmatrix} \omega - B_z \bar{u}_1 \\ 0 \end{bmatrix}. \]  

(2.25)

An auxiliary signal \( z(t) \in \mathbb{R}^3 \), in order to inject a control signal \( \bar{u}_1(t) \) into the translational dynamics from the rotational (attitude) dynamics using \( \omega(t) \), is defined as

\[ z = \omega - B_z \bar{u}_1 \]  

(2.26)

where \( B_z = [I_3, O_{3 \times 1}] \in \mathbb{R}^{3 \times 4} \). Thus, the open-loop error signals are obtained

\[ \dot{r} = \begin{bmatrix} -S(\omega) r_p + \alpha v - \alpha R^T \ddot{p}_d - R^T \dddot{p}_d + W_1 \Theta_1 \\ -\psi_d \end{bmatrix} 
+ \begin{bmatrix} -\frac{1}{m} e_p \\ 0 \end{bmatrix} + B_b \begin{bmatrix} z \\ 0 \end{bmatrix} + B_o \bar{u}_1 \]  

(2.27)

where \( B_b(\cdot) \in \mathbb{R}^{4 \times 4} \) is

\[ B_b = \begin{bmatrix} -S(\delta) & B_1 \\ T_z(\Theta) & 0 \end{bmatrix}. \]  

(2.28)

Taking the time derivative of \( z(t) \) in (3.21) and multiplying by the inertia matrix, \( J \), yields

\[ J \dot{z} = J \dot{\omega} - J B_z \ddot{u}_1. \]  

(2.29)
Substituting the second equation of (3.1) for $J\dot{\omega}(t)$ into (2.29), grouping terms, and invoking Property P1 for the linear parameterization of $N_2(\theta_2, v, |v|)$ produces

$$J\ddot{z} = -S(\omega)J\omega - Y_2(v, |v|)\theta_2 - JB_z \ddot{u}_1 + B_2u_2$$

(2.30)

where the control input $u_2(t)$ has finally appeared and it will be designed later to derive the closed-loop controller form and achieve system stability. Before designing the control input $u_2(t)$ in (3.24), the following property is stated.

**P3:** A linear parameterization $W_3\Theta_3$ is defined as

$$W_3\Theta_3 = -S(\omega)Jw - Y_2(v, |v|)\theta_2 - JB_z \ddot{u}_1$$

where $W_3(\cdot) \in \mathbb{R}^{3 \times q}$ is a known regression matrix and $\theta_3 \in \mathbb{R}^q$ is a constant parameters vector.

Using Property P3, (3.24) is rewritten as

$$J\ddot{z} = W_3(\cdot)\Theta_3 + B_2u_2.$$  

(2.31)

**Lyapunov-based Robust Control**

**Stability Analysis**

A Lyapunov analysis is used to guide the control design. The non-negative functions $V(t)$ is chosen as

$$V = \frac{1}{2}(e_p^T e_p + r^T r + z^T Jz).$$  

(2.32)

The function $V(t)$ has the following property

$$\frac{1}{2}\lambda_1 \|\eta\|^2 \leq V \leq \frac{1}{2}\lambda_2 \|\eta\|^2$$

(2.33)

where $\lambda_1 = \min\{1, \lambda_{\text{min}}(J)\}$ and $\lambda_2 = \max\{1, \lambda_{\text{max}}(J)\}$. The time derivative of $V(t)$ yields

$$\dot{V} = e_p^T \dot{e}_p + r^T \dot{r} + z^T J \ddot{z}.$$  

(2.34)
After substituting (3.13) and (3.17) into (4.71) produces
\[
\dot{V} = e^T_p \left[ \frac{1}{m} (r_p - \alpha e_p - \delta) - S(\omega) e_p + (\frac{1}{m} - 1) R^T \dot{p}_d \right] \\
+ [r^T_p, e_p] \left[ \begin{bmatrix} \alpha v - S(\omega) r_p - \alpha R^T \dot{p}_d - R^T \dot{p}_d \\ -\psi_d \end{bmatrix} \\
W_1 \Theta_1 - \frac{1}{m} e_p \right] + B_b \begin{bmatrix} z \\ 0 \end{bmatrix} + B_b \ddot{u}_1 + z^T J \dot{z} \tag{2.35}
\]

Then, the skew-symmetric terms in the first and second row can be removed. After rearranging, the equation (4.72) becomes
\[
\dot{V} = -\frac{\alpha}{m} e^T_p e_p + e^T_p \left[ (\frac{1}{m} - 1) R^T \dot{p}_d - \frac{1}{m} \delta \right] \\
+ [r^T_p, e_p] \left[ \begin{bmatrix} \alpha v - \alpha R^T \dot{p}_d - R^T \dot{p}_d \\ -\psi_d \end{bmatrix} \\
W_1 \Theta_1 \right] + B_b \begin{bmatrix} z \\ 0 \end{bmatrix} + B_b \ddot{u}_1 + z^T J \dot{z} \tag{2.36}
\]

where \( \bar{B}_b^T \in \mathbb{R}^{4 \times 3} \) is obtained from the left three columns in (2.28) and the transposition of \( \bar{B}_b(\cdot) \in \mathbb{R}^{3 \times 4} \) in the parenthesis in (2.36) is formed as
\[
\bar{B}_b^T = \begin{bmatrix} -S(\delta) \\ T_z(\Theta) \end{bmatrix}, \quad \bar{B}_b = [S(\delta), T_z^T(\Theta)] \tag{2.37}
\]

where \(-S(\delta)^T = S(\delta)\) is used. Thus, the last bracketed term in (3.17) can be rewritten as
\[
B_b \begin{bmatrix} z \\ 0 \end{bmatrix} = \bar{B}_b^T z \in \mathbb{R}^4 \tag{2.38}
\]

Translational Input Design

The control input \( \ddot{u}_1(t) \) in (3.17) can be designed based on (4.72) in the Lyapunov stability analysis as
\[
\ddot{u}_1 = B_b^{-1} (-k_r r_p + \begin{bmatrix} -\alpha v + \alpha R^T \dot{p}_d + R^T \dot{p}_d \\ \psi_d \end{bmatrix} \\
W_1 \Theta_1 \end{bmatrix} - \begin{bmatrix} \rho \|\dot{e}\| \\ \varepsilon_1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix}) = B_b^{-1} U \tag{2.39}
\]

where \( k_r = \text{diag}(k_{r1}, k_{r1}, k_{r1}, k_{r2}) \in \mathbb{R}^{4 \times 4} \) is a positive constant gain matrix, the term \(-k_r r(t) \in \mathbb{R}^4 \) is a feedback term designed to promote the convergence of \( r(t) \) to zero,
the measurable signals in the first bracket of the second row of (4.72) can be directly canceled, and other estimated terms in $W_1 \hat{\Theta}_1$ can be obtained as

$$W_1 \hat{\Theta}_1 = G(\hat{m}g) - Y_1(v, |v|)\hat{\theta}_1 + \frac{1}{m} e_p$$ (2.40)

where $\hat{\Theta}_1 \in \mathbb{R}^p$ is the unknown constant parameter vector. The last term, $\rho_1^2 (\|\zeta_1\|_s) \frac{r_p}{\varepsilon_1}$, where $\zeta_1 = [ p^T \quad p_d^T \quad r_p^T ]^T$, is a robust term to compensate for parameter mismatch term in (2.46), the bounding function $\rho_1 (\|\zeta_1\|)$ is defined as

$$\rho_1 (\|\zeta_1\|_m) \leq \rho_1 (\|\zeta_1\|) \leq \rho_1 (\|\zeta_1\|_s) \quad (2.41)$$

where the functions $\|\zeta_1\|_m$ and $\|\zeta_1\|_s$ are defined in the following manner

$$\|\zeta_1\|_s = \sqrt{\zeta_1^T \zeta_1 + \sigma}$$
$$\|\zeta_1\|_m = \sqrt{\zeta_1^T \zeta_1 + \sigma} - \sqrt{\sigma} = \|\zeta_1\|_s - \sqrt{\sigma} \quad (2.42)$$

in which $\sigma \in \mathbb{R}^1$ represents a small positive constant and $\|\zeta_1\|_m \leq \|\zeta_1\| \leq \|\zeta_1\|_s$.

Finally, the closed-loop filtered tracking error dynamics for $\dot{r}(t)$ is formed from the translational input design by substituting (2.39) into (3.17) to yield

$$\dot{r} = W_1 \hat{\Theta}_1 - S(\omega) r_p - \frac{e_p}{m} - \rho_1^2 (\|\zeta_1\|_s) r_p \quad (2.43)$$

where the parameter mismatch term $\hat{\Theta}_1(\cdot) \in \mathbb{R}^p$ is introduced as follows

$$\hat{\Theta}_1 = \Theta_1 - \hat{\Theta}_1 \quad (2.44)$$

where

$$\hat{\Theta}_1 = [ \hat{m}g \quad \hat{\theta}_1 \quad \hat{m}_2 ]^T,$$

in which $\hat{m}g = mg - \hat{m}g$ and $\hat{m}_2 = \frac{1}{m} - \frac{1}{\hat{m}}$, and the parameter mismatch term is upper bounded by

$$\|W_1 \hat{\Theta}_1\| \leq \rho_1 (\|\zeta_1\|). \quad (2.45)$$

Thus, substituting (2.39) into (2.36) yields

$$\dot{V} = -\alpha m_e^T e_p + e_p^T \left[ (\frac{1}{m} - 1)R_d \hat{p}_d - \frac{\hat{g}}{m} \right] + [r_T, e_\psi]
\left( \begin{bmatrix} W_1 \hat{\Theta}_1 \\ 0 \end{bmatrix} - k_r r - \frac{\rho_1^2 (\|\zeta_1\|_s)}{\varepsilon_1} r_p \right) + \bar{B}_b^T z + z^T J_\dot{z}. \quad (2.46)$$
Torque Input Design

The control input \( u_2(t) \in \mathbb{R}^3 \) is now formulated from (2.31) in the following form

\[
u_2 = B_2^{-1}(-k_z z - W_3(p, R, v, \omega) \hat{\Theta}_3 - \frac{\rho_2^2(\|\zeta_2\|)}{\varepsilon_2} z - \dot{B}_b r) \tag{2.47}\]

where the control input \( u_2(t) \) would be designed to stabilize the \( z(t) \)-dynamics using the feedback term \( z(t) \) and finally \( \omega(t) \) rotational dynamics and in order to compensate \( W_3(\cdot) \hat{\Theta}_3 \), in a similar way to \( W_1(\cdot) \hat{\Theta}_1 \) in (2.40), \( W_3(\cdot) \hat{\Theta}_3 \) is given by

\[
W_3 \hat{\Theta}_3 = -S(\omega) \hat{J} \omega - Y_2(v, |v|) \hat{\theta}_2 - JB_z \dot{\bar{u}}_1 \in \mathbb{R}^3 \tag{2.48}\]

where \( W_3 \in \mathbb{R}^{3 \times q} \) and \( \hat{\Theta}_3(\cdot) \in \mathbb{R}^q \) is a regression vector updated by on-line parameter estimation, the robust term \( \rho_2^2(\|\zeta_2\|) \frac{\rho}{\varepsilon_2} \), where \( \zeta_2 = [p^T \ v^T \ \omega^T]^T \), the bounding function \( \rho_2(\|\zeta_2\|) \) are defined in a similar manner as in (2.39), and the last term is added for canceling the opposite term. It is clear that \( Y_2(v, |v|) \hat{\theta}_2 \) terms was already parameterized and \( S(\omega) \hat{J} \omega(t) \) can be easily linearly parameterized. The parameterizable model for \( \bar{u}_1(t) \) needs to be developed. The time derivative of \( \bar{u}_1(t) \) can be calculated using the estimated definition in (2.39) as follows

\[
\dot{\bar{u}}_1 = \frac{d}{dt}(B_b^{-1}U) + (B_b^{-1}) \frac{d}{dt}U \tag{2.49}\]

where \( U(t) \) is the parenthetical term on the right equation as mentioned in (2.39) and the time derivative of \( U(t) \) is calculated as

\[
\frac{d}{dt}U = \begin{bmatrix}
-\alpha \dot{v} - \dot{W}_1(v) \hat{\Theta}_1 - \frac{d}{dt}(\frac{\rho_2^2(\|\zeta_1\|)}{\varepsilon_1} r_p) \\
-\alpha S(\omega) R^T \dot{p}_d + (\alpha - S(\omega)) R^T \dot{p}_d + R^T \ddot{p}_d - \psi_d
\end{bmatrix} - k_r \dot{r} \tag{2.50}\]

where \( \dot{v} \) (\( t \)) will be substituted from (3.1) including unknown parameters as

\[
\dot{v} = -S(\omega)v + \frac{1}{m}(W_1 \hat{\Theta}_1 + B_1 u_1), \tag{2.51}\]

and the time derivative of \( W_1(\cdot) \) in (2.50) is defined as

\[
\dot{W}_1 \hat{\Theta}_1 = \dot{G}(\dot{m}, \dot{g}) - \dot{Y}_1(v) \hat{\theta}_1 + \frac{1}{m} \dot{e}_p, \tag{2.52}\]
and \( \hat{e}_p(t) \) is same as (4.44) and the time derivative of the robust term related to \( \rho_1(\Vert \zeta_1 \Vert_s) \) is developed in [11]. \( \dot{r}(t) \) in (2.43) uses \( r(t) \) but \( r_p(t) \) can be linearly divided up into known and unknown terms each as

\[
r = \begin{bmatrix} \alpha e_p + \delta - R^T \hat{p}_d \\ e_\psi \end{bmatrix} + \begin{bmatrix} mv \\ 0 \end{bmatrix},
\]

from the definition of (2.15) by substituting (2.10)

\[
r_p = \alpha e_p + \delta + mv - R^T \hat{p}_d,
\]

the last term in (2.50) yields via (2.43)

\[
\dot{r} = -k_r \left( \begin{bmatrix} mv \\ 0 \end{bmatrix} + \begin{bmatrix} \alpha e_p + \delta - R^T \hat{p}_d \\ e_\psi \end{bmatrix} \right) + \bar{B}_b^T z
\]

\[
+ \begin{bmatrix} W_1 \tilde{\Theta}_1 - S(\omega)r_p - \frac{e_\psi}{m} - \frac{\rho_2^2(\Vert \zeta_1 \Vert_s)}{\varepsilon_2} r_p \end{bmatrix}.
\]

Finally, (2.49) can now be implemented using (2.43), (2.51), (2.52), (2.55), (4.44), and the time derivative of the robust term, \( \rho_1(\Vert \zeta_1 \Vert_s) \), to produce parameterization as follows

\[
\dot{u}_1 = \Phi_1 + \Phi_2 \hat{\Theta}_2
\]

where \( \Phi_1(\cdot) \in \mathbb{R}^3, \Phi_2 \in \mathbb{R}^{3 \times l} \) are a signal vector and matrix without having unknown parameters where \( \hat{\Theta}_2 \in \mathbb{R}^l \), is an unknown parameter vector. After substituting (3.32) into (2.31) and then rearranging the equation, we have the final form for the closed-loop system as shown below

\[
J \dot{z} = -k_z z + W_3 \tilde{\Theta}_3 - \bar{B}_b^T r - \frac{\rho_2^2(\Vert \zeta_2 \Vert_s)}{\varepsilon_2} z
\]

where the regression estimation error, \( \tilde{\Theta}_3(\cdot) \), is defined as \( \tilde{\Theta}_3 = \Theta_3 - \hat{\Theta}_3 \) and

\[
\left\| W_3 \tilde{\Theta}_3 \right\| \leq \rho_2 \left( \Vert \zeta_2 \Vert \right).
\]

The first bracketed term in (2.46) can be upper bounded by \( \delta_1 \) in the following manner

\[
\left\| \frac{(1 - m) R^T \hat{p}_d - \delta}{m} \right\| \leq \delta_1
\]
where \( \delta_1 \in \mathbb{R}^1 \) is a constant. Using the Young’s inequality, the first bracketed term can be further upper bounded as

\[
\| e_p \| \delta_1 \leq \frac{1}{2} \left( \varepsilon_3 \| e_p \|^2 + \frac{1}{\varepsilon_3} \delta_1^2 \right) \tag{2.60}
\]

where \( \varepsilon_3 \in \mathbb{R}^1 \) is a positive constant. After substituting (2.39), (2.57) by (3.32), and (2.60) into (2.46), the following terms, \( r_p^T \tilde{B}_b z \) and \( z^T \tilde{B}_b^T r_p \), are then canceled each other and then \( \dot{V}(t) \) produces the following upper boundedness

\[
\dot{V} \leq -\left( \frac{\alpha}{m} - \frac{\varepsilon_3}{2} \right) \| e_p \|^2 - \lambda_{\text{min}} \{ k_r \} \| r \|^2 - \lambda_{\text{min}} \{ k_z \} \cdot \\
\| z \|^2 + \| r_p^T \| \left| W_1 \tilde{\Theta}_1 \right| - \frac{\rho_1^2 (\| \zeta_1 \|)}{\varepsilon_1} \| r_p \|^2 \\
+ \| z^T \| \left| W_3 \tilde{\Theta}_3 \right| - \frac{\rho_2^2 (\| \zeta_2 \|)}{\varepsilon_2} \| z \|^2 + \frac{\delta_1^2}{2\varepsilon_3}. \tag{2.61}
\]

Using (2.45) and (2.58), the following upper bound yields

\[
\| r_p^T \| \left| W_1 \tilde{\Theta}_1 \right| - \frac{\rho_1^2 (\| \zeta_1 \|)}{\varepsilon_1} \| r_p \|^2 \\
\leq \| r_p^T \| \rho_1 (\| \zeta_1 \|) - \frac{\rho_1^2 (\| \zeta_1 \|)}{\varepsilon_1} \| r_p \|^2, \tag{2.62}
\]

\[
\leq -\frac{1}{\varepsilon_1} \left( \rho_1 (\| \zeta_1 \|) \| r_p \| - \frac{\varepsilon_1}{2} \right)^2 + \frac{\varepsilon_1}{4} \leq \frac{\varepsilon_1}{4},
\]

and also the following is obtained in a similar manner as

\[
\| z^T \| \left| W_3 \tilde{\Theta}_3 \right| - \frac{\rho_2^2 (\| \zeta_2 \|)}{\varepsilon_2} \| z \|^2 \\
\leq \| z^T \| \rho_2 (\| \zeta_2 \|) - \frac{\rho_2^2 (\| \zeta_2 \|)}{\varepsilon_2} \| z \|^2 \\
\leq -\frac{1}{\varepsilon_2} \left( \rho_2 (\| \zeta_2 \|) \| z \| - \frac{\varepsilon_2}{2} \right)^2 + \frac{\varepsilon_2}{4} \leq \frac{\varepsilon_2}{4} \tag{2.63}
\]

and then, specifying upper bounds by combining the above constant terms yields

\[
\varepsilon_0 > \frac{\delta_1^2}{2\varepsilon_3} + \frac{\varepsilon_1}{4} + \frac{\varepsilon_2}{4}. \tag{2.64}
\]

Then, upper bound can be written as

\[
\dot{V} \leq -\gamma \| \eta \|^2 + \varepsilon_0 \tag{2.65}
\]

where a positive constant \( \gamma \) is given by

\[
\gamma = \min\left\{ \frac{\alpha}{m} - \frac{1}{2\lambda_1}, \lambda_{\text{min}} \{ k_r \}, \lambda_{\text{min}} \{ k_z \} \right\}. \tag{2.66}
\]
Thus, $\dot{V}$ can be upper bounded by utilizing (2.33) as

$$\dot{V} \leq -\frac{2\gamma}{\lambda_2} V + \varepsilon_0,$$

(2.67)

using the upper boundedness of $V$ by $||\eta||^2$ in (2.33) as $||\eta||^2 \geq \frac{2}{\lambda_2} V$. Solve the inequality (2.67) to obtain

$$V(t) \leq V(0) \exp \left( -\frac{2\gamma}{\lambda_2} t \right) + \frac{\lambda_2 \varepsilon_0}{2\gamma} \left( 1 - \exp \left( -\frac{2\gamma}{\lambda_2} t \right) \right),$$

where $V(0) \leq \frac{1}{2} \lambda_2 ||\eta(0)||^2$. Therefore,

$$V(t) \leq \frac{\lambda_2}{2} ||\eta(0)||^2 + \frac{\lambda_2 \varepsilon_0}{2\gamma},$$

where $V(t)$ is exponentially driven into a ball with a radius of $\varepsilon_0$ from the term $\frac{\lambda_2 \varepsilon_0}{2\gamma}$.

The tracking result yields

$$||\eta(t)|| \leq \sqrt{\frac{2}{\lambda_2} V(t)} = \sqrt{\frac{\lambda_2}{\lambda_1} ||\eta(0)||^2 \exp \left( -\frac{2\gamma}{\lambda_2} t \right) + \frac{\lambda_2 \varepsilon_0}{2\gamma} \left( 1 - \exp \left( -\frac{2\gamma}{\lambda_2} t \right) \right)},$$

(2.68)

provided the following control gain condition holds

$$\alpha > \frac{\varepsilon_3}{2} m.$$

**Theorem 1:** The control laws given in (2.39) and (3.32), respectively ensure that the tracking error is globally uniformly ultimately bounded as

$$||\eta(t)|| \leq \sqrt{\frac{\lambda_2}{\lambda_1} ||\eta(0)||^2 e^{-\frac{2\gamma}{\lambda_2} t} + \frac{\lambda_2 \varepsilon_0}{2\gamma} \left( 1 - e^{-\frac{2\gamma}{\lambda_2} t} \right)}$$

(2.69)

where $\eta(t) \in \mathbb{R}^{10}$ is defined as

$$\eta = [\varepsilon^T, r^T, z^T]^T,$$

(2.70)

provided the following control gain condition holds

$$\alpha > \frac{\varepsilon_3 m}{2}.$$

**Remark:** Based on Theorem 1, it can be shown that all signals remain bounded in the closed-loop system.
Simulation and Implementation

Figures and Simulation

To simulate this controller, the computer system is configured to run QNX Real-Time Operating System (RTOS) and host the QMotor [12] control and simulation package. A QMotor program was written to simulate the rigid body kinematics dynamics and the feedback control. Using the yaw, pitch, roll representation results in a singularity at \( \theta_{\text{pitch}}(t) = \pm \frac{\pi}{2} \). The solution to this problem is to avoid \( \pm \frac{\pi}{2} \). The equations are then integrated on both sides using an Adams Integrator at a 1000 Hz update frequency. The tracking control in the presence of uncertainty was simulated using reference values of a small quad-rotor unmanned aerial vehicle as

\[
m = 1.2 \, [kg], \quad g = 9.81[ms^2] \quad \text{and} \quad J = \text{diag}(0.40, 0.40, 0.60) \, [kg \cdot m^2].
\]

An aerodynamic model including the aerodynamic forces and moments required to describe \( \tilde{N}(v, \omega) \) in (3.1) is

\[
Y_1(v, |v|) = \begin{bmatrix}
||v||_s v_1 & 0 & 0 \\
0 & ||v||_s v_2 & 0 \\
0 & 0 & ||v||_s v_3
\end{bmatrix}
\]

\[
Y_2(v, |v|) = \begin{bmatrix}
||v||_s v_2 & 0 & 0 \\
0 & ||v||_s v_1 & 0 \\
0 & 0 & 0
\end{bmatrix}
\]

\[
\theta_1 = [Cd_1, Cd_1, Cd_2]^T, \quad \theta_2 = [Cd_4, Cd_3, 0]^T
\]

where the coefficients are given as

\[
Cd_1 = 0.065, \quad Cd_2 = 0.065, \quad Cd_3 = 0.065.
\]

The time derivative of \( Y_1(v, |v|)\theta_1 \) used in (2.52) was implemented as follows

\[
\dot{Y}_1\theta_1 = \text{diag} \{ ||v||_s v_1 + ||v|| \dot{v}_1, ||v||_s v_2 + ||v|| \dot{v}_2, ||v||_s v_3 + ||v|| \dot{v}_3 \} \theta_1
\]

where

\[
||v||_s = \sqrt{v_1^2 + v_2^2 + v_3^2 + \sigma_s}, \quad \sigma_s \in \mathbb{R}^+,
\]
and the time derivative of vector norm yields
\[ ||\dot{v}||_s = \frac{v^T}{||v||_s} \dot{v}. \]  
(2.75)

The estimation terms and regression matrices in (2.39) and (3.32) which satisfies the linear parameterization were developed in this simulation as
\[
W_1 = \begin{bmatrix} R^T E_z & Y_1(v,|v|) & e_p \end{bmatrix} \in \mathbb{R}^{3 \times 7},
\]
(2.76)
\[
\hat{\Theta}_1 = \begin{bmatrix} \hat{m}g & \hat{\theta}_1^T & \frac{1}{\hat{m}} \end{bmatrix}^T \in \mathbb{R}^7.
\]
(2.77)

and
\[
W_3(\cdot) \in \mathbb{R}^{3 \times 41}, \Theta_3 \in \mathbb{R}^{41}
\]
(2.78)
where the members of the regression matrix in \(W_3(\cdot)\) and estimation terms \(\Theta_3\) are given [11] as
\[
(p, \Theta, \hat{\Theta}, \hat{\Theta}, R, v, \omega, r_p, e_{\psi}, z, p_d, \dot{p}_d, \ddot{p}_d, \dot{\psi}_d, \ddot{\psi}_d). \quad (2.79)
\]

The desired position and yaw trajectory are given in the following form
\[
\begin{align*}
p_d(t) &= \begin{bmatrix} p_{dx} \\ p_{dy} \\ p_{dz} \end{bmatrix} = \begin{bmatrix} \sin\left(\frac{1}{10}t\right)(1-e^{-0.1t^2}) \\ \sin\left(\frac{1}{10}t\right)(1-e^{-0.1t^2}) \\ \sin\left(\frac{1}{10}t\right)(1-e^{-0.1t^2}) \end{bmatrix} (m), \\
\psi_d(t) &= \sin(\pi t) \text{ (rad)}. \quad (2.80)
\end{align*}
\]

The constant control parameters for controller were empirically chosen to be
\[
k_{r_1} = 5, \ k_{r_2} = 5, \ k_z = 5, \ \alpha = 75, \ \delta_3 = -1,
\]
\[
\sigma_{z_1} = 1, \ \sigma_{z_2} = 1, \ \varepsilon_1 = 1, \ \text{and} \ \varepsilon_2 = 1
\]

Figure 2.1 shows the position tracking of the quad-rotor to the desired trajectory in three dimension. The actual quad-rotor trajectory represented by the blue line follows the desired trajectory represented by the red line. Figure 2.2 shows the position and yaw tracking at each axis and Figure 2.3 shows corresponding tracking error about the coordinates \((x, y, z)\) and yaw angle. Figure 2.4 shows the control inputs.
Figure 2.1 Robust Tracking Demonstration

Figure 2.2 Position \((x, y, z)\) and Yaw Angle \((\psi)\) Tracking at Each Axis
Figure 2.3 Tracking Errors in Position \((x, y, z)\) and Yaw Angle \((\psi)\) at the Each Axis

Figure 2.4 Force about Z-axis and Torques about Roll, Pitch, and Yaw
Conclusion

This chapter described designing a state feedback controller for tracking control of a quad-rotor UAV system considering parametric uncertainties has been achieved. The parameter uncertainty is compensated by the robust terms designed by a Lyapunov-based control approach. A backstepping approach is introduced to the highly non-linear coupled and cascaded system. The controller is then designed for position tracking in the three linear dimensions and the yaw angle A global uniformly ultimately bounded (GUUB) tracking result is achieved using Lyapunov-type stability analysis.


CHAPTER 3
OUTPUT FEEDBACK TRACKING CONTROL

Introduction

Recently advances in unmanned aerial vehicle construction, sensors, digital electronics, and control design have seen a rapid increase in unmanned aerial vehicles (UAV) use. The potential for UAVs in applications as diverse as emergency response, environmental monitoring, or surveillance has been well established. One particularly interesting small aerial vehicle that seems to have benefited from these technology developments is the quadrotor UAV.

The quadrotor helicopter consists of four independently driven sets of rotors that are coordinated to provide lift in the vertical direction and rotations about the roll, pitch, and yaw axes. This configuration can only directly produce a thrust along the z-axis (vertical) and torques about the roll, pitch, and yaw axes; hence, the system is underactuated and the translational motion in the x- and y-directions is indirectly achieved. An important property of the quadrotor system is that it can be modeled as coupled translational and angular subdynamics where the output of the rotational dynamics are cascaded as the input to the translational dynamics. This coupling between translational and angular dynamic subsystems arises from the gyroscopic effects, which are typically neglected in a hovering model. A feasible control solution to a system in this form is the backstepping approach, the reader is referred to [13] where the control of general cascaded dynamic is addressed. The work in [5] presented the results of two model-based control techniques applied to an underactuated quadrotor dynamic model for hovering and vertical takeoff and landing (VTOL). In [10], a backstepping approach to control a specific model of a quadrotor, the X4 flyer, is presented. This work includes the dynamic complication of the aerodynamic and gyroscopic effects of the rotating blades. The work in [6] presented attitude stabilization of the quadrotor aircraft using the backstepping technique. In [16], the authors
proposed a backstepping control suited for position and yaw tracking of a nonlinear underactuated quadrotor dynamic by leveraging the interconnected subsystems model.

Many emerging technologies use vision systems (camera-based) to estimate positions to measure motion. Camera systems may be vehicle based and used to estimate changes in scenery or may be ground-based to monitor a UAV in a fixed area. The work given in [7] is representative of the vehicle based vision applications for landing. The work in [21] also demonstrates the use of a vision system to estimate the positions and angles of a quadrotor during indoor flight. The work in [2] proposed a pose estimation algorithm for controlling a quadrotor helicopter using dual visual feedback. Although angular velocity is the fundamental measurement in angular measurement sensors such as gyroscopes, it must be derived in many vision-based systems and may be noisy as a result. If a trend were to be predicted based on review of literature, it would be that linear and angular positions will be more accurately attained as technology evolves. If this trend is true, then a controller that uses only position and angle information is well motivated. In order to meet the control objectives using only position and angle signals, without measuring velocities or accelerations, an estimate of the unmeasurable quantities is sought. In [6], the authors make a systematic presentation of the observed integrator backstepping technique for joint velocities in an n-link robot. Recently [23] presented a method for attitude stabilization and landing of a small VTOL UAV using an observer without measuring velocity and the paper by [17] proposed an observation strategy to estimate the orientation of a UAV.

A simple dynamic model, such as a linearized model or hovering model, may not account for the effects of large attitude motion, commands which could be required to follow a complicated trajectory. The work in [9] presented a trajectory tracking controller for an underactuated small helicopter using a backstepping procedure. The authors in [19] proposed a trajectory tracking control for unmanned air vehicles with constrained velocity and heading rate. In [11], the authors designed and developed a miniature flight control system and created a multi-vehicle agent platform. The authors in [1] proposed a solution of the trajectory-tracking and path-following for
underactuated autonomous vehicles. The work in [20] presented control strategies for a quadrotor system capable of automatic VTOL, hovering, and obstacle avoidance using simple minimal sensing.

In this section, a tracking controller for desired 3D position and yaw commands is designed using the nonlinear dynamic model of the quadrotor helicopter and only output feedback signals. The yaw angle controller will provide an extra degree of freedom for tasks such as surveillance or object targeting. An observer, velocity estimator, is designed to compensate for the unmeasurable signals. The observer and controller are designed concurrently via Lyapunov stability arguments in order to ensure a stability result [12]. The choice here to control the three linear translations requires that some of the desired translational commands be “reformulated” as attitude commands via the backstepping approach. An integrator backstepping method in which the virtual force command is backstepped through the integrator to the rotational controller is introduced in order to achieve the tracking control objectives. The novelty of this work is that we propose a trajectory tracking control approach that uses only position and angle measurements and is fully supported by a mathematical stability proof. The payoff from this approach is the mathematical assertion of semi-global, uniformly ultimately bounded tracking while obtaining the control objectives.

System Model

Quad-Rotor Aerial Vehicle

The translational and rotational dynamic equations of motion in the body-fixed reference frame, $B$, are based on [12]

$$
\begin{bmatrix}
  m I_3 & O_{3 \times 3} \\
  O_{3 \times 3} & J
\end{bmatrix}
\begin{bmatrix}
  \dot{v} \\
  \dot{\omega}
\end{bmatrix}
= \begin{bmatrix}
  -m S(\omega) & O_{3 \times 3} \\
  O_{3 \times 3} & S(J\omega)
\end{bmatrix}
\begin{bmatrix}
  v \\
  \omega
\end{bmatrix}
- \begin{bmatrix}
  N_1(\theta_1, v, |v|) \\
  N_2(\theta_2, \omega, |\omega|)
\end{bmatrix}
+ \begin{bmatrix}
  G_1(R) \\
  O_{3 \times 1}
\end{bmatrix}
+ \begin{bmatrix}
  B_1 \\
  O_{3 \times 1}
\end{bmatrix}
\begin{bmatrix}
  u_1 \\
  u_2
\end{bmatrix}
$$

where $v(t) \in \mathbb{R}^3$ denotes the linear velocity, $\omega(t) \in \mathbb{R}^3$ represents the angular velocity, $m \in \mathbb{R}^1$ is the known mass of the quad-rotor, $J \in \mathbb{R}^{3 \times 3}$ denotes a positive definite inertia matrix, $G_1(R) \in \mathbb{R}^3$ is a gravity vector, and $N_1(\theta_1, v, |v|)$, $N_2(\theta_2, \omega, |\omega|) \in \mathbb{R}^3$
are the aerodynamic damping interactions where $\theta_1, \theta_2 \in \mathbb{R}^3$ are damping coefficients. The input $u_1(t) \in \mathbb{R}^1$ provides lifting force in the $z$-direction and $u_2(t) \in \mathbb{R}^3$ creates torque in the roll, pitch, and yaw directions. The construction of the quad-rotor links the inputs to the dynamics via $B_1 = [0, 0, 1]^T \in \mathbb{R}^3$ and $B_2 = I_3 \in \mathbb{R}^{3 \times 3}$. Additionally, $I_3$ is an identity matrix, $O_{n \times m}$ is a zero matrix, and $S(\cdot) \in \mathbb{R}^{3 \times 3}$ is a general skew-symmetric matrix.

The Euler based angle, $\Theta(t) = [\phi, \theta, \psi]^T \in \mathbb{R}^3$ for roll, pitch, and yaw, rotation matrix $R(\Theta) \in SO(3)$ \cite{22} translates a vector in the body-fixed frame into inertial coordinates and the angular velocity transformation matrix, $T(\Theta) \in \mathbb{R}^{3 \times 3}$, is used to relate the rate of change in the Euler angles to the angular velocities in the body-fixed frame \cite{8}. The gravity vector is represented in the body-frame using $R^T(\Theta) = mgR^T[0, 0, 1]^T \in \mathbb{R}^3$ where the unit vector in the coordinates of the inertial frame is used and $g \in \mathbb{R}^1$ denotes the acceleration due to gravity. The translational and rotational kinematic equations in the body-fixed reference frame are given by

$$\begin{bmatrix} v^T, \omega^T \end{bmatrix} = D(R, T)\dot{x} \in \mathbb{R}^6,$$

$$\dot{x} = \begin{bmatrix} \dot{p}^T, \Theta^T \end{bmatrix} \in \mathbb{R}^6 \tag{3.2}$$

where $D(R, T) = \text{diag}(R^T(\Theta), T^{-1}(\Theta)) \in \mathbb{R}^{6 \times 6}$, $x = [p^T, \Theta^T]^T \in \mathbb{R}^6$, and $p(t) \in \mathbb{R}^3$ contains the position of the body-fixed reference frame relative to the inertial frame and $\dot{p}(t) \in \mathbb{R}^3$ represents the translational velocity in the inertial frame. In order to transform (3.1) into the inertial frame for subsequent control development, (3.1) can be rewritten

The dynamics in (3.1) can be rewritten by substituting from (3.2) for $[v^T, \omega^T]^T$ in (3.1) yielding

$$M[v^T, \omega^T]^T = CD(R, T)\dot{x} - N_b(\theta_1, \theta_2, |v|, |\omega|, v, \omega) + G(R) + B\bar{U} \tag{3.3}$$

where

$$M = \begin{bmatrix} mI_3 & O_3 \\ O_3 & J \end{bmatrix} \in \mathbb{R}^{6 \times 6},$$

$$C = \begin{bmatrix} -mS(\omega) & O_3 \\ O_3 & S(J \omega) \end{bmatrix} \in \mathbb{R}^{6 \times 6},$$

and the definitions of $N_b(\cdot) = \begin{bmatrix} N_1^T(\theta_1, v, |v|) \\ N_2^T(\theta_2, \omega, |\omega|) \end{bmatrix} \in \mathbb{R}^6$, $G(R) \in \mathbb{R}^6$, $B \in \mathbb{R}^{6 \times 4}$, and $\bar{U} \in \mathbb{R}^4$ are apparent from the operation. The time derivative of $[v^T, \omega^T]^T$
in (3.3) can be related to \( \ddot{x}(t) \) by differentiating (3.2) to yield

\[
[\dot{v}^T, \dot{\omega}^T]^T = \frac{d}{dt} (D(R, T)) \dot{x} + D(R, T) \ddot{x} = \ddot{D}(R, T, \dot{x}) \ddot{x} + D(R, T) \dddot{x}
\]  

(3.4)

where

\[
\frac{d}{dt} (D(R, T)) \triangleq \ddot{D}(R, T, \dot{x}) = \left[ \frac{d}{dt}(R^T), \frac{d}{dt}O_3, \frac{d}{dt}(T^{-1}(\Theta)) \right] \in \mathbb{R}^{6 \times 6},
\]

\[
\frac{d}{dt}(R^T) = -S(\omega)R^T, \text{ and } \frac{d}{dt}(T^{-1}(\Theta))|_{\dot{\Theta}} = \frac{\partial}{\partial \Theta}(T^{-1}(\Theta))\dot{\Theta} \in \mathbb{R}^{3 \times 3} \text{ where } \frac{\partial}{\partial \Theta}(T^{-1}(\Theta)) \in \mathbb{R}^{3 \times 3 \times 3} \text{ is a tensor. Multiplying (3.4) by } M(\cdot), \text{ substituting into (3.3) for } M[\dot{v}^T, \dot{\omega}^T]^T, \text{ and arranging terms yields}
\]

\[
MD(R, T) \ddot{x} = CD(R, T) \ddot{x} - M\dddot{D}(R, T, \dot{x}) \ddot{x} - N_b(\theta_1, \theta_2, |v|, |\omega|, v, \omega) + G(R) + B\dot{U}.
\]

(3.5)

Premultiplying (3.5) by \( D^\top(R, T) \) transforms the dynamic equation into the inertial frame for subsequent control development, compacting the result yields the dynamic model

\[
\ddot{M}(T) \ddot{x} = \ddot{C}(T, \dot{x}) \ddot{x} - \dddot{N}(\theta_1, \theta_2, R, T, |\dot{x}|, \dot{x}) + \dddot{G} + \dot{B}(R, T) \ddot{U}
\]

(3.6)

where \( \ddot{M}(T) = D^\top M D(R, T) \in \mathbb{R}^{6 \times 6} \) denotes the inertia matrix, \( \ddot{C}(T, \dot{x}) = D^\top(R, T) \cdot (C D(R, T) - M \dddot{D}(R, T, \dot{x})) \in \mathbb{R}^{6 \times 6} \) is a Coriolis-centrifugal force matrix, \( \dddot{N}(\theta_1, \theta_2, R, T, |\dot{x}|, \dot{x}) = D^\top(R, T) N_b(\theta_1, \theta_2, |v|, |\omega|, v, \omega) \in \mathbb{R}^{6} \) is an aerodynamic damping term, \( \dddot{G} = D^\top(R, T) G(R) \in \mathbb{R}^{6} \) is a gravity term, and \( \ddot{B}(R, T) = D^\top(R, T) B \in \mathbb{R}^{6 \times 4} \) represents the input matrix, \( \frac{d}{dt}(D(R, T)) = \dot{D}(R, T, \dot{x}) = \text{diag} \left( \frac{d}{dt}(R^T), \frac{d}{dt}(T^{-1}(\Theta)) \right) \in \mathbb{R}^{6 \times 6} \), in which \( \frac{d}{dt}(R^T) = \dot{R}^T = -S(\omega)R^T \), and \( \frac{d}{dt}(T^{-1}(\Theta))|_{\dot{\Theta}} = \frac{\partial}{\partial \Theta}(T^{-1}(\Theta))\dot{\Theta} \in \mathbb{R}^{3 \times 3} \) where \( \frac{\partial}{\partial \Theta}(T^{-1}(\Theta)) \in \mathbb{R}^{3 \times 3 \times 3} \) is a tensor.

Model: Properties and Assumption

The dynamic system given in (3.6) satisfies the following properties

Property 1: \( \ddot{M}(T) \) and \( \ddot{C}(T, \dot{x}) \) are skew-symmetric property in the sense \( \xi^T \left( \frac{d}{dt}(\ddot{M}(T)) \right) + 2\ddot{C}(T, \dot{x}) \xi = 0, \ \forall \xi \in \mathbb{R}^{6} \).

Property 2: The inertia matrix \( \ddot{M}(T) \) is symmetric, positive-definite, and can be upper and lower bounded as \( \lambda_1 \| \xi \|^2 \leq \xi^T \ddot{M}(T) \xi \leq \lambda_2 \| \xi \|^2, \ \forall \xi \in \mathbb{R}^{6} \) where \( \lambda_1, \lambda_2 \in \mathbb{R}^{1} \) are positive constants.
The reader is referred to [14] for proof of the properties. The following assumptions are made regarding the dynamic model.

**Assumption 1**: \( \theta \neq \pm \frac{\pi}{2} \) or \( T^{-1}(\Theta) \) exists and \( \|T(\Theta)\|_{\infty} \leq \varepsilon_t \) where \( \varepsilon_t \in \mathbb{R}^1 \) is a positive constant.

**Assumption 2**: Only position and angle are measurable. The desired trajectories are bounded; i.e., \( p_d(t), \dot{p}_d(t), \ddot{p}_d(t), \dot{\psi}_d(t), \ddot{\psi}_d(t) \in L_{\infty} \).

**Assumption 3**: The aerodynamic terms \( N_1(\theta_1, |v|, v) \) and \( N_2(\theta_2, |\omega|, \omega) \in \mathbb{R}^3 \) in (3.1) can be represented in the following forms

\[
N_1(\theta_1, |v|, v) \equiv Y_1(\theta_1, |v|)v \quad \text{and} \quad N_2(\theta_2, |\omega|, \omega) \equiv Y_2(\theta_2, |\omega|)\omega
\]

where \( Y_1(\theta_1, |v|) \) and \( Y_2(\theta_2, |\omega|) \in \mathbb{R}^{3 \times 3} \) are known regression matrices. These terms expressed in the inertial frame, \( \tilde{N}(\theta_1, \theta_2, R, T, |\dot{x}|, \dot{x}) \) in (3.6), can be expressed as

\[
\tilde{N}(\theta_1, \theta_2, R, T, |\dot{x}|, \dot{x}) \equiv Y_n(\theta_1, \theta_2, R, T, |\dot{x}|)\dot{x}
\]

where \( Y_n(\theta_1, \theta_2, R, T, |\dot{x}|) \in \mathbb{R}^{6 \times 6} \). Additionally, the following bounds are made for subsequent stability analysis: \( Y_1(\theta_1, |v|) \leq \xi_{a1} \|v\| \quad \text{and} \quad Y_2(\theta_2, |\omega|) \leq \xi_{a2} \|\omega\| \), and \( Y_n(\theta_1, \theta_2, R, T, |\dot{x}|) \leq \xi_{e1} \|\dot{x}\| \) where \( \xi_{a1}, \xi_{a2}, \) and \( \xi_{e1} \) are positive constants.

Output Feedback Tracking Controller Design

An observer and closed-loop controller are designed based on the tracking error dynamics of the underactuated UAV system. Roughly speaking, the observer, \( \dot{x}(t) \in \mathbb{R}^6 \), is designed to estimate the velocities \( v(t), \omega(t) \), which are not measurable, by using inputs and outputs of the UAV and the dynamic model of the UAV in (3.6). The control inputs are then designed to force the system to track the desired position and yaw angle based on the estimated velocities while ensuring stability.

**Observer Design**

The first step is to create an estimate of the unmeasurable states \( \dot{v}(t) \) and \( \dot{\omega}(t) \in \mathbb{R}^3 \) by creating the signal \( \dot{x}(t) \) such that

\[
[\dot{v}^T, \dot{\omega}^T]^T = D(R, T) \dot{x}, \quad \dot{x} = [\dot{\theta}^T, \dot{\Theta}^T]^T. \tag{3.7}
\]
The signal \( \dot{x} (t) \) is generated [6] as follows
\[
\dot{x} = \dot{y} + k_1 \ddot{x}
\] (3.8)
where the initial condition is \( \dot{y}(0) = -k_1 \ddot{x}(0), k_1 \in \mathbb{R}^1 \) is a positive gain, and \( \ddot{x}(t) = x - \ddot{x} = [\ddot{p}^T, \ddot{\Theta}^T] \in \mathbb{R}^6 \). The auxiliary signal \( \dot{y}(t) \in \mathbb{R}^6 \) introduced in (3.8) is updated according to
\[
\dot{y} = \tilde{M}^{-1}(\tilde{C}(T, \dot{x}_o)\dot{x}_o - \tilde{N}(\theta_1, \theta_2, R, T, |\dot{x}_o|, \dot{x}_o) + \tilde{G} + \tilde{B}U + k_2 \ddot{x}) + k_3 \beta \ddot{x}
\] (3.9)
where \( k_2, k_3, \) and \( \beta \in \mathbb{R}^1 \) are positive constants and an auxiliary signal \( \dot{x}_o(t) = \dot{x} - \beta \ddot{x} \in \mathbb{R}^6 \) is used. The basic form of the internal observer signal \( \dot{y}(t) \in \mathbb{R}^6 \) comes from substituting \( \dot{x}_o(t) \) into the modeling equation of (3.6) and adding the \( \ddot{x} \)-terms to promote stability of the observer. A filtered observer error signal is defined [3] as
\[
s = \dot{x} - \dot{x}_o = \ddot{x} + \beta \ddot{x} \in \mathbb{R}^6,
\] (3.10)
in which the velocity estimation error is defined as \( \ddot{x}(t) = \dot{x} - \ddot{x} \in \mathbb{R}^6 \) where
\[
[\ddot{p}^T, \ddot{\Theta}^T]^T = D(R, T)\ddot{x}, \ddot{p}(t) = \dot{p} - \dot{\hat{p}}, \text{ and } \ddot{\Theta}(t) = \dot{\Theta} - \dot{\hat{\Theta}}.
\]
The filtered observer error dynamics are obtained from the derivative of (3.10) by defining \( \ddot{x} = \dddot{x} - \dot{x} \), and then using (3.6) and the time derivative of (3.8). Initial substitutions yield
\[
\tilde{M}(T)s = \tilde{C}(T, \dot{x})s + \tilde{C}(T, \dot{x})\dot{x}_o - \tilde{C}(T, \dot{x}_o)\dot{x}_o - Y_n(\theta_1, \theta_2, R, T, |\dot{x}|)\dot{x} + (3.11)
\]
\[
Y_n(\theta_1, \theta_2, R, T, |\dot{x}_o|)\dot{x}_o - [k_2 + k_3 \beta \tilde{M}(T)]\ddot{x} + [\beta - k_01] \tilde{M}(T)\dddot{x}
\]
where the definition of \( s(t) \) was substituted as \( \ddot{x} = s + \dot{x}_o \) and Assumption 3 was utilized by adding and subtracting \( Y_n(\theta_1, \theta_2, R, T, |\dot{x}|)\dot{x}_o \).

Property 3: The Coriolis-centrifugal terms satisfy [14]
\[
\tilde{C}(T, \dot{x})\dot{x}_o - \tilde{C}(T, \dot{x}_o)\dot{x}_o = \tilde{C}_e(T, \dot{x}_o)s.
\]
Letting \( k_{01} = \beta + k_3 \), utilizing Assumption 3 and Property 3, and then substituting (3.10) yields the filtered velocity observer error dynamics
\[
\tilde{M}(T)s = \tilde{C}(T, \dot{x})s + \tilde{C}_e(T, \dot{x}_o)s - Y_n(\theta_1, \theta_2, R, T, |\dot{x}|)s +
\]
\[
Y_n(\theta_1, \theta_2, R, T, |\dot{x}| - |\dot{x}_o|)\dot{x}_o - k_2 \ddot{x} - k_3 \tilde{M}(T)s.
\] (3.12)
FSFB Error Systems

In order to demonstrate the approach to designing an OFB controller the same control design approach is demonstrated for the less complicated full-state feedback (FSFB) error systems, i.e., we temporarily assume that the velocities are measurable in order to highlight the backstepping control design approach. The position tracking error, \( e_p(t) = R^T(p - p_d) \in \mathbb{R}^3 \), is defined as the transformed difference between the inertial-frame based position, \( p(t) \), and the inertial-frame based desired position, \( p_d(t) \in \mathbb{R}^3 \). The position tracking error rate, \( \dot{e}_p(t) \in \mathbb{R}^3 \), is obtained by taking the time derivative of \( e_p(t) \) and substituting \( \dot{R}^T = -S(\omega)R^T \), substituting \( \dot{p}(t) = Rv \) from (3.2), using \( R^TR = I_3 \) and the definition of \( e_p(t) \) to collect terms, and then adding and subtracting \( \frac{1}{m}R^T\dot{p}_d \). The translational velocity tracking error, \( e_v(t) = mv - R^T\dot{p}_d \in \mathbb{R}^3 \), is defined in the body-fixed frame to create the open-loop position tracking error dynamics as follows

\[
\dot{e}_p = -S(\omega)e_p + \frac{1}{m}e_v + \left( \frac{1}{m} - 1 \right)R^T\dot{p}_d. \tag{3.13}
\]

Taking the time derivative of \( e_v(t) \), substituting for \( m\dot{v}(t) \) from (3.1) and \( \dot{R}^T = -S(\omega)R^T \), and applying the definition of \( e_v(t) \) yields the velocity error rate

\[
\dot{e}_v = -S(\omega)e_v + G_1(R) - Y_1(\theta_1,|v|)v - R^T\dot{p}_d + B_1u_1 \tag{3.14}
\]

where Assumption 3 was invoked to parameterize \( N_1(\theta_1,|v|,v) \). The yaw angle tracking error is defined as \( e_\psi(t) = \psi - \psi_d \in \mathbb{R}^1 \). The yaw angle rate error system is derived by taking the time derivative of \( e_\psi(t) \) as follows

\[
\dot{e}_\psi = \dot{\psi} - \dot{\psi}_d = T_z(\Theta)\omega - \dot{\psi}_d \tag{3.15}
\]

where the term \( T_z(\Theta) \in \mathbb{R}^{3\times3} \) is the third row of \( T(\Theta) \). Note that \( T_z(\Theta)\omega = \dot{\psi} \) and \( \dot{\psi}_d \) is the desired yaw angle rate in the inertial frame. In order to further develop the control design, the filtered position tracking error signal \( r_p(t) = e_v + \alpha e_p + \delta \in \mathbb{R}^3 \) is defined in which \( \delta = [0,0,\delta_3]^T \in \mathbb{R}^3 \) is a constant design vector and \( \delta_3, \alpha \in \mathbb{R}^1 \).
are constants. The filtered position tracking error can be combined with the yaw tracking error to create a composite filtered tracking error \( r(t) = [r_p^T, e_\psi]^T \in \mathbb{R}^4 \).

The composite filtered tracking error dynamics can be found by differentiating \( r(t) \) to yield

\[
\dot{r}(t) = [\dot{r}_p^T, \dot{e}_\psi]^T = [\dot{e}_\theta^T + \alpha \dot{e}_p^T, \dot{e}_\psi]^T \in \mathbb{R}^4.
\]

The filtered position tracking error rate, \( \dot{r}_p(t) \), is obtained by substituting (3.13) and (3.14), and then adding and subtracting the term \( S(\omega)\delta \) to facilitate the introduction of \( r_p(t) \) on the right-hand side as

\[
\dot{r}_p = \frac{\alpha}{m} e_v - S(\omega) r_p + (\frac{\alpha}{m} - \alpha) R^T \dot{p}_d - R^T \ddot{p}_d + G_1(R) - Y_1(\theta_1, |v|) v + S(\omega) \delta + B_1 u_1. \tag{3.16}
\]

It is now a straightforward matter to substitute from (3.15) and (3.16) into \( \dot{r}(t) \) to yield the open-loop filtered tracking error dynamics in the following form

\[
\dot{r} = \begin{bmatrix}
\frac{\alpha}{m} e_v - S(\omega) r_p + (\frac{\alpha}{m} - \alpha) R^T \dot{p}_d - R^T \ddot{p}_d + G_1(R) - Y_1(\theta_1, |v|) v \\
-\dot{\psi}_d
\end{bmatrix}
+ \begin{bmatrix}
-S(\delta) \\
T_z(\Theta)
\end{bmatrix}
\begin{bmatrix}
\omega \\
u_1
\end{bmatrix}
\tag{3.17}
\]

where \( S(\omega)\delta = -S(\delta)\omega \) was used [8]. The last term will be referred to as \( B_\mu \mu \) where \( B_\mu \in \mathbb{R}^{4 \times 4} \) and \( \mu \in \mathbb{R}^4 \).

**Backstepping Approach**

The last term in (3.17) highlights the location where the control input will be eventually designed. Note that \( \mu(t) \) contains of two elements that can affect the translational dynamics. One component, thrust \( u_1(t) \), is a physical input to the system that can be directly specified in a control strategy. The second component, the angular velocity \( \omega(t) \), can be thought of as the output of the rotational dynamic subsystem (the lower three rows in (3.1)) that can only be indirectly specified through control of the rotational dynamics via the system input \( u_2(t) \). Thus, the integrator backstepping approach is pursued as the means of controlling the translational dynamics via the angular velocity \( \omega(t) \). The control command \( \bar{u}_1(t) \in \mathbb{R}^4 \) can be introduced into the composite filtered error dynamics by adding and subtracting \( B_\mu \bar{u}_1(t) \) into (3.17) [12] to yield

\[
\dot{r} = \begin{bmatrix}
\frac{\alpha}{m} e_v - S(\omega) r_p + (\frac{\alpha}{m} - \alpha) R^T \dot{p}_d - R^T \ddot{p}_d + G_1(R) - Y_1(\theta_1, |v|) v \\
-\dot{\psi}_d
\end{bmatrix}
+ B_\mu \bar{u}_1 + B_\mu(\mu - \bar{u}_1).
\tag{3.18}
\]
Now, two matrices $B_z = [I_3, O_{3	imes1}] \in \mathbb{R}^{3\times4}$ and $B_o = [0, 0, 0, 1]^T \in \mathbb{R}^{1\times4}$ are created to separate $\bar{u}_1(t)$ in the last parenthetical term in (3.18) as

$$\mu - \bar{u}_1 = \begin{bmatrix} (\omega - B_z \bar{u}_1)^T, (u_1 - B_o \bar{u}_1)^T \end{bmatrix}^T.$$

(3.19)

The last operation makes it clear that the physical input $u_1(t)$ will be designed directly through the design of $\bar{u}_1(t)$ in the sense that the thrust input $u_1(t)$ is designed using $u_1 = B_o \bar{u}_1$ which yields

$$\mu - \bar{u}_1 = \begin{bmatrix} (\omega - B_z \bar{u}_1)^T, 0 \end{bmatrix}^T.$$

(3.20)

The result in (3.20) illustrates that the last term in (3.18) is in fact an error term that can be controlled through the input to the rotational dynamics, to this end $z(t) \in \mathbb{R}^3$ is defined to quantify the closeness of the control term $\bar{u}_1(t)$ to the angular velocities $\omega(t)$ as

$$z = \omega - B_z \bar{u}_1.$$

(3.21)

Thus, arranging (3.18) using (3.20) and (3.21) yields

$$\dot{r} = \begin{bmatrix} \frac{\alpha \omega}{m} - S(\omega) r_p + \frac{\alpha(1-m)}{m} R^T \dot{p}_d - R^T \dot{p}_d + G_1(R) - Y_1(\theta_1, |v|) v \end{bmatrix} + B_\mu(\bar{u}_1 + \begin{bmatrix} z \\ 0 \end{bmatrix}).$$

(3.22)

The effect of introducing the control signal $\bar{u}_1(t)$ through the manipulations in (3.18) - (3.22) was to create a desired output of the rotational dynamics. The new control signal $\bar{u}_1(t)$ can be designed to control all three translational axes; but this injected control signal then becomes the tracking objectives for the rotational dynamics. The design now proceeds to ensure that the auxiliary signal $z(t)$ in (3.21) is driven to a small value. Taking the time derivative of $z(t)$ in (3.21), multiplying by the inertia matrix, and substituting the second equation of (3.1) for $J \dot{\omega}(t)$ produces

$$J \ddot{z} = S(J \omega) \omega - N_2(\theta_2, \omega, |\omega|) + B_2 u_2 - J B_z \dot{u}_1.$$

(3.23)

It is now useful to group terms in equation (3.23) and invoke Assumption 3 for the parameterization of $N_2(\omega)$ as

$$J \ddot{z} = Y_3(\theta_2, \theta_3, \dot{u}_1, |\omega|, \omega) + B_2 u_2,$$

(3.24)
in which the linear parameterization $Y_3(\theta_2, \theta_3, \dot{u}_1, |\omega|, \omega) = S(J\omega)\omega - Y_2(\theta_2, |\omega|)\omega - JB_x \dot{u}_1$ is made where $Y_3(\theta_2, \theta_3, \dot{u}_1, |\omega|, \omega) \in \mathbb{R}^3$ and $\theta_3 \in \mathbb{R}^3$ is a inertia parameters vector. Equation (3.24) now contains the rotational subsystem inputs $u_2(t)$. If our intention were to actually design the full-state feedback controller, the error systems in (3.13), (3.14), (3.15), (3.22), and (3.24) would be incorporated into a Lyapunov-type analysis from which $u_1(t)$ and $u_2(t)$ would be specified.

Output Feedback Controller Formulation

The output feedback controller is now formulated using the components outlined in the previous sections, i.e., the observer, the approach to error system development, and the remapping of the rotational inputs to achieve translation objectives are modified for this purpose. In the OFB only the position tracking error, $e_p(t)$, and yaw angle tracking error, $e_\psi(t)$, are measurable.

Translational Input Design

The filtered tracking error dynamics in (3.22) presents the opportunity to design the auxiliary control input $\bar{u}_1(t)$. A term of the form $-r(t) \in \mathbb{R}^4$ will be required to promote the convergence of $r(t)$ to zero; however, $r(t)$ is not measurable so the signal $\hat{r}(t) \in \mathbb{R}^4$ is introduced to represent the observed filtered tracking error defined by

$$\hat{r}(t) = [\hat{r}_p^T, e_\psi]^T = [\dot{\hat{e}}_v^T + \alpha \hat{e}_p^T + \dot{\delta}^T, \psi - \dot{\psi}_d]^T.$$ 

The component definition of $\hat{r}(t)$ can be differentiated to yield

$$\dot{\hat{r}}(t) = [\ddot{\hat{e}}_v^T + \alpha \dot{\hat{e}}_p^T, \dot{\psi} - \ddot{\psi}_d]^T.$$ 

The control $\bar{u}_1(t)$ in (3.22) can be redesigned as

$$\bar{u}_1 = B_\mu^{-1}(-k_r \hat{r} + \begin{bmatrix} R^T \dot{\hat{p}}_d - \frac{\alpha \dot{\hat{e}}_v}{m} - \frac{\alpha(1-m)}{m} R^T \ddot{\hat{p}}_d - G_1(R) + Y_1(\theta_1, |\dot{\psi}|)\dot{\psi} - \frac{e_\psi}{m} \end{bmatrix}) \quad (3.25)$$

where $k_r = \text{diag}(k_{r1}, k_{r1}, k_{r2}, k_{r2}) \in \mathbb{R}^{4 \times 4}$ is a positive constant matrix. Note that the last term, $\frac{e_\psi(t)}{m}$, is designed to cancel the opposite sign during the stability analysis. Substituting (3.25) into (3.22) yields

$$\dot{r} = -k_r \hat{r} + \begin{bmatrix} \frac{\alpha(e_v - \dot{\hat{e}}_v)}{m} - S(\omega)r_p - Y_1(\theta_1, |v|)v - Y_1(\theta_1, |\dot{\psi}|)\dot{\psi} - \frac{e_\psi}{m} \end{bmatrix} + B_\mu \begin{bmatrix} z \\ 0 \end{bmatrix}. \quad (3.26)$$

The observed filtered tracking error, $\hat{r}(t)$, can be defined by subtracting $\hat{r}(t)$ from $r(t)$ as

$$\hat{r}(t) = r - \hat{r} = [\hat{r}_p^T, 0]^T = [m \dot{v}^T, 0]^T \in \mathbb{R}^4.$$
as \( \tilde{e}_v(t) = e_v - \hat{e}_v = m\dot{v} - \dot{\tilde{r}}_p \) where \( \ddot{v}(t) = v - \dot{v} \) is the velocity tracking error. This information can be used in equation (3.26) by substituting \(-r(t) + \hat{r}(t)\) for \(-\dot{r}(t)\) and substituting \(-e_v(t) + \hat{e}_v(t)\) for \(-\dot{e}_v(t)\) to yield the closed-loop filtered tracking error dynamics in the following form

\[
\dot{r} = -k_r r + k_r \begin{bmatrix} \hat{r}_p \\ 0 \end{bmatrix} + \begin{bmatrix} \frac{a}{m} \hat{e}_v - S(\omega)r_p - \hat{Y}_1(\theta_1, \dot{\hat{\psi}}, \dot{\hat{v}}, v) - \frac{a_p}{m} \\ 0 \end{bmatrix} + B_\mu \begin{bmatrix} z \\ 0 \end{bmatrix},
\]

(3.27)

where \( \hat{Y}_1(\theta_1, \dot{\hat{\psi}}, \dot{\hat{v}}, v) = Y_1(\theta_1, |\dot{v}|)v - Y_1(\theta_1, |\dot{v}|)\dot{v} \in \mathbb{R}^{3 \times 3} \) is introduced.

**Output Feedback Error Systems Formulation**

Looking to the design of \( u_2(t) \), we can see from (3.23) that the design of \( u_2(t) \) will need to compensate for \( \dot{\hat{u}}_1(t) \). An observed velocity tracking error signal can be defined as \( \hat{e}_v(t) = m\ddot{v} - R^T \hat{p}_d \in \mathbb{R}^3 \), the velocity error rate equation in (3.14) is redefined by substituting \( \dot{\hat{v}}(t) \) and \( \hat{\dot{\omega}}(t) \) to yield the observed velocity rate error as

\[\hat{e}_v = -S(\omega)\dot{e}_v + G_1(R) - Y_1(\theta_1, |\dot{v}|)\dot{v} - R^T \hat{p}_d + B_1 u_1.\]

In a similar fashion, \( \hat{e}_p(t) \in \mathbb{R}^3 \) can be formed from (3.13) by substituting \( \hat{\dot{\omega}}(t) \) and \( \hat{e}_v(t) \) as

\[\hat{e}_p = -S(\dot{\omega})e_p + \frac{1}{m} \hat{e}_v + (\frac{1}{m} - 1)R^T \hat{p}_d.\]

The observed filtered position error rate, \( \dot{\hat{r}}_p(t) \), is found by substituting \( \hat{e}_v(t) \) and \( \hat{e}_p(t) \) into the top row of \( \hat{\dot{r}}(t) \) to yield

\[\dot{\hat{r}}_p = -S(\dot{\omega})\dot{\hat{r}}_p + \frac{a}{m} \hat{\dot{e}}_v + \frac{a(1-m)}{m} R^T \hat{p}_d + G_1(R) - Y_1(\theta_1, |\dot{v}|)\dot{v} - R^T \hat{p}_d - S(\delta)\hat{\dot{\omega}} + B_1 u_1.\]

(3.28)

The observed yaw error rate is found by substituting \( \hat{\dot{\omega}}(t) \) into (3.15) as \( \dot{\hat{\psi}} = \psi - \dot{\psi}_d = T_z(\Theta)\dot{\omega} - \dot{\psi}_d. \) The observed tracking error rate is then obtained by substituting \( \dot{\hat{\psi}}(t) \) and (3.28) into \( \dot{\hat{r}}(t) \) as

\[\dot{\hat{r}} = \begin{bmatrix} \frac{a}{m} \hat{e}_v - S(\dot{\omega})\dot{\hat{r}}_p + \frac{a(1-m)}{m} R^T \hat{p}_d - R^T \hat{p}_d + G_1(R) - Y_1(\theta_1, |\dot{v}|)\dot{v} \\ -\dot{\psi}_d \end{bmatrix} + B_\mu (\ddot{\hat{u}}_1 + \begin{bmatrix} \ddot{z} \\ 0 \end{bmatrix}),\]

(3.29)

where the auxiliary signal \( \ddot{z}(t) \in \mathbb{R}^3 \) is introduced through \( \hat{\dot{\omega}}(t) \) by the definition \( \ddot{z}(t) = \dot{\omega} - B_z \ddot{\hat{u}}_1. \) The auxiliary estimation signal \( \tilde{z}(t) \in \mathbb{R}^3 \) can be defined as \( \tilde{z} = z - \ddot{z} \) but can be immediately reduced using (3.21) and the above definition of \( \ddot{z}(t) \) to \( \ddot{z}(t) = \omega - \dot{\omega}(t) = \ddot{\hat{z}}. \) The derivative of \( \ddot{z}(t) \) is \( \dddot{z} = \ddot{\omega} - B_z \ddot{\hat{u}}_1. \)

**Torque Input Design**
The FSFB development makes it clear through (3.24) that ideally the control input
\( u_2(t) \in \mathbb{R}^3 \) would be designed to stabilize the \( z(t) \)-dynamics and cancel \( Y_3(\theta_2, \theta_3, \hat{u}_1, |\omega|, \omega) \). To achieve this using measurable signals, the dynamics of the signal \( \hat{z}(t) \) are developed. Multiplying \( \hat{z}(t) \) by \( J \) yields \( J\hat{z}(t) = J\hat{\omega} - JB_z\hat{u}_1 \). The terms \( \hat{\omega}(t) \) and \( \hat{\omega}(t) \) are substituted into the second equation of (3.1) for \( \omega(t) \) and \( \hat{\omega}(t) \) and the result substituted into \( J\hat{z}(t) \) to produce

\[
J\hat{z} = S(J\hat{\omega})\hat{\omega} - N_2(\theta_2, |\omega|, \hat{\omega}) + B_2u_2 - JB_z\hat{u}_1 = Y_3(\theta_2, \theta_3, \hat{u}_1, |\omega|, \hat{\omega}) + B_2u_2. \tag{3.30}
\]

The estimated velocities provide the means to parameterize (3.30) in a manner similar to (3.24), the estimated velocities \( \hat{\omega}(t) \) and \( \hat{\omega}(t) \) are substituted into \( Y_3(\theta_2, \theta_3, \hat{u}_1, |\omega|, \omega) \) to create the estimate \( Y_3(\theta_2, \theta_3, \hat{u}_1, |\omega|, \omega) = S(J\hat{\omega})\hat{\omega} + Y_2(\theta_2, \hat{\omega})\hat{\omega} - JB_z\hat{u}_1 \) where \( Y_2(\theta, \hat{\omega}) \) and \( Y_3(\theta_2, \theta_3, \hat{u}_1, |\omega|, \omega) \) are the estimated regression matrices. The proposed control signal in (3.25) can be written as \( \hat{u}_1 = B_\mu^{-1}U \) in order to calculate the time derivative of \( \hat{u}_1(t) \) as \( \hat{u}_1(t) = \frac{\partial}{\partial t}(B_\mu^{-1}U) + (B_\mu^{-1}) \frac{\partial U}{\partial t} \). The term \( B_\mu^{-1}(\cdot) \) can be measured since \( \Theta(t) \) is measurable and \( \frac{\partial}{\partial t}(B_\mu^{-1}) \) can be estimated using \( \hat{\phi}(t) = T_x(\Theta)\hat{\omega} \) and \( \hat{\theta}(t) = T_y(\Theta)\hat{\omega} \). The time derivative of \( U(t) \) is now found to be

\[
\frac{d}{dt}U = \begin{bmatrix}
-\frac{\omega}{m} \hat{e}_v + \frac{\partial}{\partial t}(Y_1(\theta_1, |\hat{v}|)\hat{v}) - \frac{1}{m} \hat{e}_p + R^T \hat{p}_d - \alpha \left( 1 - \frac{1}{m} \right) R^T \delta_
u \\
-k_r \hat{e}_r + \frac{S(\omega)}{\psi_d} \left( \frac{\omega}{m} R^T \hat{p}_d - \alpha R^T \hat{p}_d - R^T \delta_
u + G_1(R) \right)
\end{bmatrix} \tag{3.31}
\]

where \( \hat{e}_v, \hat{e}_p, (3.29) \), and \( \dot{G}_1(R) = -S(\omega)G_1(R) \) were utilized and the time derivative of \( Y_1(\theta_1, |\hat{v}|)\hat{v} \) must still be explicitly calculated (as is done in the computer simulation). The control input \( u_2(t) \) in (3.24) is now formulated using (3.30) as a guide as

\[
u_2 = B_2^{-1} \left( -k_z \hat{z} - Y_3(\theta_2, \theta_3, \hat{u}_1, |\omega|, \omega) - B_\mu^T \hat{r} \right) \tag{3.32}
\]

where the first term is a feedback term to promote the convergence of \( z(t) \) to zero, the last term is added to cancel a similar term during the stability analysis, and \( B_\mu^T = [-S^T(\delta), T_x^T(\Theta)] \in \mathbb{R}^{3 \times 4} \). The control is now substituted into (3.24), the result is then manipulated by substituting \(-z(t) + \tilde{z}(t) \) for \( z(t) \) and using \( \tilde{\omega}(t) = \tilde{z}(t) \) for \( \hat{\omega}(t) \) to yield the closed-loop system as

\[
J\tilde{z} = -k_z z + k_z \tilde{\omega} + \tilde{Y}_3(\theta_2, \theta_3, \omega, \tilde{\omega}) - B_\mu^T r + mS(\delta)\tilde{v} \tag{3.33}
\]
where \( \tilde{r}(t) \) was used to create the last two terms, and the regression estimation error, \( \tilde{Y}_3(\theta_2, \theta_3, \omega, \dot{\omega}, \ddot{\omega}) \in \mathbb{R}^3 \), is defined as \( Y_3(\theta_2, \theta_3, \hat{u}_1, |\omega|, \omega) - Y_3(\theta_2, \theta_3, \hat{u}_1, |\omega|, \dot{\omega}) \). The skew-symmetry \( S^T(\xi) = -S(\xi) \) is invoked to rewrite the matrix \( \bar{B}_\mu^T(\cdot) \) as \( \bar{B}_\mu^T = [S(\delta), T_z^T(\Theta)] \), and hence, we have \( \bar{B}_\mu^T \tilde{r}(t) = [S(\delta), T_z^T(\Theta)] [m\bar{v}^T, 0]^T = mS(\delta)\bar{v}. \)

### Stability Analysis

It is now possible to summarize the expected tracking performance of the proposed observer/controller system. The outline of a Lyapunov-type proof is given in Appendix ?? and full details of the proof along with a boundedness check of all signals are given in [14].

**Theorem 1:** The velocity observer of (3.8) and (3.9) and the control law of (3.25) and (3.32) ensure that the tracking error is semi-globally uniformly ultimately bounded (SGUUB) in the sense that

\[
\|\eta(t)\| \leq \sqrt{\frac{\lambda_4}{\lambda_3}} \|\eta(0)\|^2 \exp(-\frac{2\lambda_5}{\lambda_4}t) + \frac{\lambda_4\varepsilon_0^2}{2\lambda_3\lambda_5\lambda_6} (1 - \exp(-\frac{2\lambda_5}{\lambda_4}t)) \quad (3.34)
\]

where \( \eta = [\varepsilon_p^T, r^T, z^T, s^T, \dot{\theta}^T]^T \), and \( \varepsilon_0, \lambda_2, \lambda_5, \) and \( \lambda_6 \) are positive constants, and \( \lambda_3, \lambda_4 \) are positive constants given by \( \lambda_3 = \min\{1, \lambda_1, k_2\} \) and \( \lambda_4 = \max\{1, \lambda_2, k_2\} \), under the parameters gain condition that

\[
k_2 > \frac{1}{\beta_3} \left[ \xi_{c_4}\sigma_3 + \xi_{f_1}\sigma_3 f_1^2(\|\eta(0)\|) + \xi_{f_2}\sigma_3 (4\varepsilon_\sigma f_1^2(\|\eta(0)\|) + 5\varepsilon_8 f_1(\|\eta(0)\|) + \varepsilon_{11})^2 \right],
\]

\[
k_3 > \frac{1}{\lambda_1} \left[ \left. \frac{\xi_{c_1} (4\varepsilon_\sigma f_1^2(\|\eta(0)\|) + 5\varepsilon_9 f_1(\|\eta(0)\|) + \varepsilon_{11}) \right. }{\xi_{c_4}\sigma_3 + \xi_{f_1}\sigma_3 f_1^2(\|\eta(0)\|) + \xi_{f_2}\sigma_3 (4\varepsilon_\sigma f_1^2(\|\eta(0)\|) + 5\varepsilon_8 f_1(\|\eta(0)\|) + \varepsilon_{11})^2} \right. \right]
\]

\( k_r > 4, k_z > 4, \) and \( \alpha > \frac{\lambda_2}{2} \mu \) where \( f_1(\cdot) \) is listed in (??), \( \xi_{c_1}, \xi_{c_4}, \xi_{f_1}, \xi_{f_2}, \sigma_3, \sigma_6, \varepsilon_0, \varepsilon_\sigma, \varepsilon_8 \) to \( \varepsilon_{10} \), and \( \varepsilon_{11} \) are all positive constants.

**Remark 1:** The SGUUB tracking result given by (3.34) can be represented by the steady state bound on \( \|\eta\| \) as

\[
\lim_{t \to \infty} \|\eta\| = \sqrt{\frac{\lambda_4\varepsilon_0^2}{2\lambda_3\lambda_5\lambda_6}}. \quad (3.35)
\]

From (3.35), it is clear that the value of \( \eta(t) \) and the steady-state bound for the tracking error \( \eta(t) \) can be arbitrarily small by decreasing the value \( \varepsilon_0 \).
Figure 3.1 Output Feedback Trajectory-Tracking Demonstration. The command and result are split into two plots to show the rise of the UAV (left) and the descent (right).

Figure 3.2 3D Position and Yaw Angle Tracking (left) and Error (right) at Each Axis.
Figure 3.3 Control inputs; the translational force input $u_1(t)$ and the torque commands $u_2(t)$

Figure 3.4 Estimated Velocity Output of the Velocity Observer
Figure 3.5 Gyroscopic Forces of the Rolling, Pitching, and Yawing
Simulation

The UAV output feedback tracking control was simulated in Simulink using the parameters of the vehicle $m = 1.2 \ [kg]$ and $J = diag(0.40, 0.40, 0.60) \ [kgm^2]$. The control parameters were iteratively chosen for the observer and controller as $k_1 = 4$, $k_2 = 4$, $k_3 = 2$, $\beta = 2$, $k_{r1} = 5$, $k_{r2} = 5$, $k_z = 5$, $\alpha = 4$. The aerodynamic damping terms $[16]$ were included in the system model of (3.1) by using $N_1(\theta_1, |v|, v) = [d_1 |v_1|, d_2 |v_2|, v_3, d_3 |v_3|]^T$ and $N_2(\theta_2, |\omega|, \omega) = [g_1 |\omega_1|, g_2 |\omega_2|, g_3 |\omega_3|, g_4 |\omega_3|]^T$. Since the models for the aerodynamic damping terms satisfy Assumption 3 they were used to create the parameterization $Y_1(\theta_1, |\hat{v}|)\dot{\hat{v}} = diag(d_1 |\hat{v}_1|, d_2 |\hat{v}_2|, d_3 |\hat{v}_3|)$ and $Y_2(\theta_2, |\hat{\omega}|)\dot{\hat{\omega}} = diag(g_1 |\hat{\omega}_1|, g_2 |\hat{\omega}_2|, g_3 |\hat{\omega}_3|) \in \mathbb{R}^{3 \times 3}$ where $\theta_1 = [d_1, d_2, d_3]^T$, and $\theta_2 = [g_1, g_2, g_3]^T$. The derivative of $Y_1(\theta_1, |\hat{v}|)\dot{\hat{v}}$ in $\hat{v}_1(t)$ of (3.31) is $\frac{dY_1(\theta_1, |\hat{v}|)}{dt} = diag(2d_1 |\hat{v}_1|, 2d_2 |\hat{v}_2|, 2d_3 |\hat{v}_3|)$ where $\frac{d}{dt} |\hat{v}| = \hat{v}sgn(\hat{v}), \hat{v}sgn(\hat{v}) = |\hat{v}|$, and the first equation of the modeling equation in (3.1) were utilized. In order to simulate the reality that the simulated quadrotor can produce only limited torque and thrust the control commands, $u_1(t)$ and $u_2(t)$, are limited by saturation functions. Figure 3.1 shows the position tracking of the quad-rotor to the desired trajectories $p_d(t)$. The actual quad-rotor trajectory represented by the solid line follows the desired trajectory represented by the dotted line which is a helix-like command that first rises to $2\,[m]$, then goes up to $2.5\,[m]$ in left plot and descends in the right plot while the yaw angle is commanded such that the centerline of the craft is tangent to the radius. The left plot in Figure 3.2 shows the position tracking for each of the coordinates $(x, y, z)$ and the yaw angle. The right plot represents the corresponding errors at each axis. In Figure 3.3, shows the control inputs; the translational force input $u_1(t)$ as the collective command and the torque commands $u_2(t)$ about roll, pitch, and yaw in the lower three plots. Figure 3.4 shows the estimated velocity output of the velocity observer. Figure 3.5 represents forces of the rolling, pitching, and yawing by gyroscopic effects.

Conclusion

In this chapter, the goal of designing an output feedback (OBF) controller for a quad-rotor UAV system has been demonstrated mathematically and via a computer
simulation. The mathematical result shows that a SGUUB tracking result is achieved. A backstepping approach was used to design the control for the cascaded, coupled, and underactuated system without performing linearization. While the output feedback control design was predicated on a hypothetical sensing system that only produces angular and linear positions, the simulation results were shown as initial validation of the proposed system and it does appear that sensors such as camera based units may provide justification of this approach. We believe this to be the first work to present such a comprehensive result for quad-rotor trajectory-tracking control which can be directly applicable to other unmanned vehicles.


CHAPTER 4
COORDINATED CONTROL

The problem of controlling the quadrotor UAV with an attached 2-link robot manipulator (UAVRM) as a single integrated “flying arm” is presented. The goal is to control the position and orientation of the end-effector. A primary challenge is to write the kinematics and dynamics for the UAVRM system so that the interaction of the two subsystems is modeled. Many results for under-water vehicles [1], [3] assume the mass of an attached manipulator and any reaction forces and torques have little effect of the motion of the vehicle. However, in a realistic UAV application a robotic arm will likely consume most of the payload capacity of the UAV, for example the payload capacity of the Draganflyer XPro is about 0.5kg and the vehicle weight is about 2kg, thus the interaction of the subsystem should be included in the model.

**UAVRM Kinematics**

In order to control the UAV and robot manipulator, the forward kinematics from the rigid-body UAV to the end-effector is derived using the Denavit-Hartenberg Convention. The UAV rigid-body coordinate frame is used as a reference as shown in Figure 4.1 and the two-link robot manipulator is serially connected from this point. The body-fixed coordinate frame is translated and rotated relative to the earth-fixed inertial frame as the UAV moves. The first joint of the two-link manipulator provides pan motion and the second joint provides pitch motion to the end effector. The combined system can be modeled as a single robotic manipulator using the Denavit-Hartenberg (D-H) table in Table 4.1.

<table>
<thead>
<tr>
<th>Link</th>
<th>$\theta_{x_{i-1}z_{i-1}}^{x_i}$ (angle)</th>
<th>$d$ (offset)</th>
<th>$a$ or $b$ (length)</th>
<th>$\alpha_{x_{i-1}x_{i-1}}^{x_i}$ (twist)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$(\psi \theta \phi) + 90^\circ$</td>
<td>$d_1$</td>
<td>0</td>
<td>$+90^\circ$</td>
</tr>
<tr>
<td>2</td>
<td>$q_1 + 90^\circ$</td>
<td>0</td>
<td>$a_2$</td>
<td>$+90^\circ$</td>
</tr>
<tr>
<td>3</td>
<td>$q_2$</td>
<td>0</td>
<td>$b_3$</td>
<td>$-90^\circ$</td>
</tr>
</tbody>
</table>

Table 4.1 Denavit-Hartenberg Convention for UAV and Robot Manipulator
The Jacobian matrix for the UAVRM can be found using

\[
\vec{J}_q = \begin{bmatrix}
    z_0 \times (o_3-o_0) & z_1 \times (o_3-o_1) & z_2 \times (o_3-o_2)
\end{bmatrix}
\]

(4.1)

\[
\equiv \begin{bmatrix}
    \vec{J}_v & \vec{J}_\omega
\end{bmatrix} \equiv \begin{bmatrix}
    \vec{J}_{v1} & \vec{J}_{v2} & \vec{J}_{v3}
    \vec{J}_{\omega1} & \vec{J}_{\omega2} & \vec{J}_{\omega3}
\end{bmatrix}.
\]

The location of the coordinate system \(o_0\) relative to the earth-fixed frame is the position, \(p_0\), of the UAV

\[
o_0 = p_0 = \begin{bmatrix} p_x & p_y & p_z \end{bmatrix}^T.
\]

(4.2)

The axis of rotation from the origin \(o_0\) is

\[
z_0 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T
\]

(4.3)

where \(z_i\) is specified in [9]. Based on the D-H table, the homogeneous matrix for the
first of the three links can be defined as

\[
A_{ZYX} = \begin{bmatrix}
R_{ZYX} & o_0 \\
O_{1\times3} & 1
\end{bmatrix}, \quad R_{z, \theta_1=+90^\circ} = \begin{bmatrix}
0 & -1 & 0 & 0 \\
1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix},
\]

\[
T_{z,d_1} = \begin{bmatrix}
I_3 & [0,0,d_1]^T \\
O_{1\times3} & 1
\end{bmatrix}, \quad \text{and} \quad R_{x, \alpha_1=+90^\circ} = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

The transformation matrix, \( R_{ZYX} \), which describes roll, pitch, and yaw rotation in order from the body-fixed frame to earth-fixed frame is developed in [3]. \( R_{ZYX} \) is the Euler-based spatial rotation matrix, \( R^I_F(\Theta) = R_{z, \psi} \cdot R_{y, \theta} \cdot R_{x, \phi} \in SO(3) \), that translates a body-fixed frame referenced quantity into inertial coordinates and is calculated from

\[
R^I_F(\Theta) = \begin{bmatrix}
c\psi c\theta & c\psi s\theta s\phi - s\psi c\phi & s\psi s\phi + c\psi c\phi s\theta \\
c\psi c\theta & c\psi c\phi s\theta + s\psi s\phi & s\theta s\psi c\phi - c\psi s\phi \\
c\psi c\phi + s\psi s\theta s\psi & s\theta s\phi c\phi - c\psi s\phi & c\theta c\phi
\end{bmatrix} (4.5)
\]

where \( \Theta(t) = [\phi, \theta, \psi]^T \in \mathbb{R}^3 \) are the Euler angles and \( c \cdot = \cos(\cdot) \) and \( s \cdot = \sin(\cdot) \) are abbreviations. The \( A_1 \) matrix can be obtained as

\[
A_1 = A_{ZYX} \cdot R_{z, \theta_1=+90^\circ} \cdot T_{z,d_1} \cdot R_{x, \alpha_1=+90^\circ} = \begin{bmatrix}
R_{z, \psi} & (s\psi s\phi + c\psi s\theta c\phi) & (s\psi s\phi + c\psi s\theta c\phi) \\
R_{y, \theta} & (s\theta s\psi c\phi - c\psi s\phi) & (s\theta s\psi c\phi - c\psi s\phi) \\
R_{x, \phi} & c\theta c\phi & c\theta c\phi
\end{bmatrix}_{01} \quad (4.6)
\]

From the \( A_1 \) matrix, the axis of rotation for joint 1, \( z_1 \), is obtained from the third column and position of this joint \( i \) coordinate relative to the earth-fixed reference frame is found from the last column of \( A_1 \) as

\[
z_1 = \begin{bmatrix}
c\psi c\theta \\
s\psi c\theta \\
-\theta
\end{bmatrix} \quad \text{and} \quad o_1 = \begin{bmatrix}
(s\psi s\phi + c\psi s\theta c\phi) d_1 \\
(s\theta s\psi c\phi - c\psi s\phi) d_1 \\
c\theta c\phi d_1
\end{bmatrix}, \quad (4.7)
\]

The transformation matrix for the second link can be defined using the values of the second row of Table 4.1 as

\[
R_{z, \theta_2=\alpha_1+90^\circ} = \begin{bmatrix}
R_{z, \theta_2=\alpha_1+90^\circ} & O_{3\times1} \\
O_{1\times3} & 1
\end{bmatrix} = \begin{bmatrix}
-sq_1 & -cq_1 & 0 & 0 \\
cq_1 & -sq_1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix},
\]
\[
T_{x,a_2} = \begin{bmatrix} I_3 & [a_2, 0, 0]^T \\ O_{1 \times 3} & 1 \end{bmatrix},
\]

\[
R_{x, a_2} = \begin{bmatrix} R_{x,a_2=+90\degree} & O_{3 \times 1} \\ O_{1 \times 3} & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c(90\degree) & -s(90\degree) & 0 \\ 0 & s(90\degree) & c(90\degree) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.
\]

Then, the \( A_{2} \) matrix can be obtained as

\[
A_{2} = R_{z, \theta_2} \cdot T_{x,a_2} \cdot R_{x, a_2} = \begin{bmatrix} -sq_1 & 0 & cq_1 & -sq_1 \cdot a_2 \\ cq_1 & 0 & sq_1 & cq_1 \cdot a_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4.8}
\]

Thus, the cumulative matrix \( T_{0}^{0} \) is obtained by post-multiplying (4.6) with (4.8) to yield

\[
T_{0}^{0} = A_{1}A_{2}. \tag{4.9}
\]

The axis of rotation, \( z_2 \), is produced by calculating (4.9) and is then obtained from the third column of the rotation matrix in \( T_{0}^{0} \), as

\[
z_2 \equiv \begin{bmatrix} z_{2x} \\ z_{2y} \\ z_{2z} \end{bmatrix} = \begin{bmatrix} (c\psi s\theta s\phi - s\psi c\phi)cq_1 + (s\psi s\phi + c\psi s\theta c\phi)sq_1 \\ (c\psi c\phi + s\psi s\theta s\phi)cq_1 + (s\psi s\theta c\phi - c\psi s\phi)sq_1 \\ (c\theta s\phi)cq_1 + (c\theta c\phi)sq_1 \end{bmatrix}. \tag{4.10}
\]

The position of joint 2 relative to the earth-fixed frame can be obtained from the last column of the matrix \( T_{0}^{0} \) as

\[
o_2 \equiv \begin{bmatrix} (s\psi c\phi - c\psi s\theta s\phi)sq_1a_2 + (s\psi s\phi + c\psi c\phi s\theta)cq_1a_2 + (s\psi s\phi + c\psi s\theta c\phi)d_1 \\ -(c\psi c\phi - s\phi s\theta s\psi)sq_1a_2 + (s\phi s\psi c\phi - c\psi s\phi)cq_1a_2 + (s\psi s\theta c\phi - c\psi s\phi)d_1 \\ -c\theta s\phi sq_1a_2 + c\theta c\phi cq_1a_2 + c\theta c\phi d_1 \end{bmatrix}. \tag{4.11}
\]

The homogeneous transformation matrix \( A_{3} \) is formed using

\[
A_{3} = R_{z, \theta_3 = a_2} \cdot T_{y,b_3} \cdot R_{x, a_3 = -90\degree} \tag{4.12}
\]
Thus, the cumulative matrix, \( T^0_{3} \), from the earth-fixed frame to the end-effector, which completes the forward kinematics, is defined as

\[
T^0_{3} = A_1A_2A_3 = \begin{bmatrix}
\psi s\theta s\phi - s\psi s\phi & s\psi s\phi + c\psi c\phi s\theta & c\psi c\theta & (s\psi s\phi + c\psi s\theta s\phi)d_1 \\
c\psi c\phi + s\phi s\theta s\psi & s\theta s\psi c\phi - c\psi s\phi & s\psi c\theta & (s\psi s\theta c\phi - c\psi s\phi)d_1 \\
c\theta s\phi & c\theta c\phi & -s\theta & c\theta c\phi d_1 \\
-sq_1 & 0 & cq_1 & -sq_1 \cdot a_2 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

The axis of rotation \( z_2 \) is obtained from the third column of the rotation matrix in \( T^0_{3} \) after evaluating (4.13) as

\[
z_3 \equiv \begin{bmatrix}
z_{3x} \\
z_{3y} \\
z_{3z}
\end{bmatrix} = \begin{bmatrix}
-[-(c\psi s\theta s\phi - s\psi s\phi)sq_1 + (s\psi s\phi + c\psi c\phi s\theta)cq_1]sq_2 + c\psi c\theta cq_2 \\
-[-(c\phi c\psi + s\phi s\theta s\psi)sq_1 + (s\theta s\psi c\phi - c\psi s\phi)cq_1]sq_2 + s\psi c\theta sq_2 \\
-[-c\theta s\phi sq_1 + c\theta c\phi cq_1]sq_2 - s\theta sq_2
\end{bmatrix},
\]

and the position of this coordinate frame relative to the earth-fixed frame can be obtained from the last column of the matrix \( T^0_{3} \) as

\[
o_3 \equiv \begin{bmatrix}
-[-(c\psi s\theta s\phi - s\psi s\phi)sq_1 + (s\psi s\phi + c\psi c\phi s\theta)cq_1]sq_2 b_3 + c\psi c\theta cq_2 b_3 \\
-
[-(c\psi s\theta s\phi - s\psi s\phi)sq_1 a_2 + (s\psi s\phi + c\psi c\phi s\theta)cq_1 a_2 + (s\psi s\phi + c\psi c\phi s\theta)d_1] \\
-[-(c\phi c\psi + s\phi s\theta s\psi)sq_1 + (s\theta s\psi c\phi - c\psi s\phi)cq_1]sq_2 b_3 + s\psi c\theta b_3 \\
-[-(c\psi c\psi + s\phi s\theta s\psi)aq_1 a_2 + (s\theta s\psi c\phi - c\psi s\phi)cq_1 a_2 + (s\theta s\psi c\phi - c\psi s\phi)d_1] \\
-[-c\theta c\phi sq_1 + c\theta c\phi cq_1]sq_2 b_3 - s\theta sq_2 b_3 - c\theta s\phi sq_1 a_2 + c\theta c\phi a_2 + c\theta c\phi d_1
\end{bmatrix}.
\]

Note that \( o_3 \) is the position of the end-effector in the earth-fixed frame and will be denoted as \( o_3 = p_E \). For \( \vec{J}_{e1} \) and \( \vec{J}_{o2} \), The first term in the Jacobian matrix \( J_q \) can be calculated by forming the link direction vector as...
For the second term in \( \bar{J} \), the cross-product is found as

\[
\begin{align*}
03 - o1 &= \begin{bmatrix}
-[cψθsφ - sψcφ]sq_1 + (sψsφ + cψcφsθ)cq_1 | sq_2b_3 + cψcθcq_2b_3 \\
-(cψθsφ - sψcφ)q_1o_2 + (sψsφ + cψcφsθ)cq_1o_2 \\
-(cψθcφ + sψθsψ)q_1o_2 + (sψsφ - cψcφ)sθcq_1o_2 \\
-(cθsφdq_1 + cθcθcq_1)q_3b_3 - sθcq_2b_3 - cθsφdq_1o_2 + cθcθcq_1o_2
\end{bmatrix}
\end{align*}
\]

and then

\[
\begin{align*}
\bar{J}_{v2} = z_1 \times (o3-o1) &= \begin{bmatrix}
\dot{x} & \dot{y} & \dot{z} \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
(o3-o1)_x \\
(o3-o1)_y \\
(o3-o1)_z
\end{bmatrix}
\end{align*}
\]

\[
\begin{align*}
\bar{J}_{v2} = z_1 \times (o3-o1) &= \begin{bmatrix}
\dot{x} & \dot{y} & \dot{z} \\
\psi cθ & sψ cθ & -sθ
\end{bmatrix}
\begin{bmatrix}
(o3-o1)_x \\
(o3-o1)_y \\
(o3-o1)_z
\end{bmatrix}
\end{align*}
\]

(4.17)

The final link direction vector is found using

\[
\begin{align*}
03 - o2 &= \begin{bmatrix}
-[cψθsφ - sψcφ]sq_1 + (sψsφ + cψcφsθ)cq_1 | sq_2b_3 + cψcθcq_2b_3 \\
-(cψθsφ - sψcφ)q_1o_2 + (sψsφ + cψcφsθ)cq_1o_2 \\
-(cψθcφ + sψθsψ)q_1o_2 + (sψsφ - cψcφ)sθcq_1o_2 \\
-(cθsφdq_1 + cθcθcq_1)q_3b_3 - sθcq_2b_3 - cθsφdq_1o_2 + cθcθcq_1o_2
\end{bmatrix}
\end{align*}
\]

and the cross-product is found to be

\[
\begin{align*}
\bar{J}_{v3} = z_2 \times (o3-o2) &= \begin{bmatrix}
\dot{x} & \dot{y} & \dot{z} \\
2_x & 2_y & 2_z
\end{bmatrix}
\begin{bmatrix}
(o3-o2)_x \\
(o3-o2)_y \\
(o3-o2)_z
\end{bmatrix}
\end{align*}
\]

(4.18)

The above components are assembled according to (4.1) to yield

\[
\bar{J}_q = \begin{bmatrix}
-(o3-o0)_y & sψcθ(o3-o1)_z + sθ(o3-o1)_y & z_{2y}(o3-o2)_z - z_{2z}(o3-o2)_y \\
-(o3-o0)_x & -sψcθ(o3-o1)_z - sθ(o3-o1)_y & z_{2z}(o3-o2)_x - z_{2x}(o3-o2)_z \\
0 & cψcθ(o3-o1)_y - sψcθ(o3-o1)_x & z_{2x}(o3-o2)_y - z_{2y}(o3-o2)_x
\end{bmatrix}
\]

(4.19)
UAVRM End-Effector Kinematics

First, the kinematics of the UAV alone can be written as

\[
\begin{bmatrix}
\dot{p} \\
\dot{\Theta}
\end{bmatrix} = \begin{bmatrix}
R_I^T(\Theta) & O_{3 \times 3} \\
O_{3 \times 3} & T_I(\Theta)
\end{bmatrix} \begin{bmatrix}
v \\
\omega
\end{bmatrix} \in \mathbb{R}^6,
\]

which can be solved for the linear velocity and angular velocity of the UAV as

\[
\begin{bmatrix}
v \\
\omega
\end{bmatrix} = \begin{bmatrix}
(R_I)^T & O_{3 \times 3} \\
O_{3 \times 3} & (T_I)^{-1}
\end{bmatrix} \begin{bmatrix}
\dot{p} \\
\dot{\Theta}
\end{bmatrix} \equiv D\dot{x} \in \mathbb{R}^6
\]

where \(v(t), \omega(t) \in \mathbb{R}^3\) denote the linear velocity and the angular velocity, respectively. \(R_I^T(\Theta)\) was given in (4.5). The body-fixed angular velocities are transformed by the angular orientation matrix \(T(\Theta) \in \mathbb{R}^{3 \times 3}\), into the inertial frame and is given by

\[
T_I^T(\Theta) = \begin{bmatrix}
T_x(\Theta) \\
T_y(\Theta) \\
T_z(\Theta)
\end{bmatrix} = \begin{bmatrix}
1 & s\phi t\theta & c\phi t\theta \\
0 & c\phi & -s\phi \\
0 & s\phi/c\theta & c\phi/c\theta
\end{bmatrix}
\]

where \(t = \tan(\cdot)\) is used.

Next, the velocity of the end-effector of the 2-link robot manipulator for alone, before attachment to the UAV-body, can be represented from an the origin \(O_0\) (here we will assume that \(O_0\) is the UAV body-fixed frame \(F\) (or denoted as \(B\) in Appendix C) using the end-effector velocity of the robot manipulator, that is measured in \(E\), (refer to the dynamic modeling of the two-link robot manipulator in Appendix B)

\[
v_{E}^F = J_v \dot{q} \\
\omega_{E}^F = J_w \dot{q}
\]

where \(J_v, J_w \in \mathbb{R}^{3 \times 2}\) are the Jacobian matrices, \(\dot{q} \in \mathbb{R}^2\) is a vector of joint velocities, and \(v_{E}^F, \omega_{E}^F \in \mathbb{R}^3\) are end-effector velocities relative to UAV body-fixed frame \(F\) and expressed in \(F\). Now consider that the UAV and the robot manipulator are attached; then the position, denoted as \(p_E\), and orientation, \(\Theta_E\), of the end-effector expressed in earth-fixed frame can be written as

\[
x_E = \begin{bmatrix}
p_E \\
\Theta_E
\end{bmatrix} \in \mathbb{R}^6.
\]

The position rate of the end-effector can be described by separating the contributing velocity components and rotational matrices, and can be decomposed into two terms;
a velocity from the earth-fixed frame to the UAV frame and a velocity from the body-fixed frame to the end-effector frame as

\[ v_{IE}^E = v_{IF}^E + v_{FE}^E, \]  

(4.25)

in which the velocities can also be represented into new terms using rotation matrices as follows:

\[ v_{IF}^E = R_{IF}^E v_{IF}^F = R_{IF}^E v^F \]  

and \( v_{FE}^E = R_{FE}^E v_{FE}^E \).

\[ v^E = v_{IF}^E, \ v^F = v_{IF}^F \]  

yields

\[ \dot{p}_E = R_{IF}^E v_{IF}^E = R_{IF}^E (v_{IF}^E + v_{FE}^E) = R_{IF}^E R_{IF}^E v_{IF}^F + R_{IF}^E R_{IF}^E v_{FE}^E = R_{IF}^E v^F + R_{IF}^E v_{FE}^F. \]  

(4.26)

Utilizing (4.23) to rewrite the last term in (4.26) yields

\[ \dot{p}_E = R_{IF}^E v^F + R_{IF}^E J_v \dot{q}. \]  

(4.27)

In order to integrate the UAV and robot manipulator, the last term in the right side of (4.27) can be redefined by modeling the forward kinematics of the integrated form as

\[ R_{IF}^E J_v \dot{q} = \bar{J}_v \dot{q} \]  

(4.28)

where \( \bar{J}_v \) represents the Jacobian transformation matrix of the robot joint velocities through the UAV rotation, \( R_{IF}^E \), into end-effector velocities in the earth-fixed inertial frame. Thus, (4.27) yields

\[ \dot{p}_E = R_{IF}^E v^F + \bar{J}_v \dot{q}. \]  

(4.29)

In a similar manner, the angular rate of the end-effector can be given by

\[ \dot{\Omega}_E = T_{IF}^E \omega_{IE}^E = T_{IF}^E R_{IE}^E \omega_{IE}^E = T_{IF}^E R_{IE}^E (\omega_{IF}^E + \omega_{FE}^E) = T_{IF}^E \omega^F + T_{IF}^E \omega_{FE}^E \]  

(4.30)

where \( \omega_{IF}^E = R_{IF}^E \omega_{IE}^E \) in the second equality, the end-effector angular velocity can be decomposed into two terms in the third equality, and then \( \omega_{IF}^E = R_{IF}^E \omega_{IF}^E = R_{IF}^E \omega^E \), and \( R_{IF}^E R_{IF}^E = I_3 \) which leads to the final form of the right-hand side of (4.30). The last in (4.30) yields by utilizing the second equation of (4.23)

\[ T_{IF}^E \omega_{FE}^E = T_{IF}^E R_{IE}^E \omega_{FE}^E = T_{IF}^E J_\omega \dot{q} = \bar{J}_\omega \dot{q} \]  

(4.31)
where $J_\omega$ is the pure Jacobian matrix between the end-effector of the 2-link robot manipulator and the origin of UAV body-fixed frame (not the earth-fixed inertial frame) as shown in Appendix C and $\bar{J}_\omega$ is the transformation matrix of the joint velocities from earth-fixed inertial frame to the end-effector coordinate frame. Substituting (4.31) into (4.30) produces

$$\dot{\Theta}_E = T_F^I \omega^F + \bar{J}_\omega \dot{q}$$

(4.32)

where $\omega^F = \omega^F_I$. Hence, the end-effector kinematics of the integrated system can be defined as

$$\dot{x}_E = \begin{bmatrix} \dot{p}_E \\ \dot{\Theta}_E \end{bmatrix} = \begin{bmatrix} R(\Theta) & O_{3\times3} \\ O_{3\times3} & T(\Theta) \end{bmatrix} \begin{bmatrix} \dot{v} \\ \dot{\omega} \\ \dot{\theta} \end{bmatrix}$$

$$\equiv D_E \xi.$$  

(4.33)

and

$$\xi = D_E^\dagger \dot{x}_E$$  

(4.34)

where $D_E^\dagger$ is the Moore-Penrose pseudo-inverse given

$$D_E^\dagger = (D_E^T D_E)^{-1} D_E^T \in \mathbb{R}^{8 \times 6}$$  

(4.35)

### Integrated System Modeling

The dynamic modeling of the integrated vehicle system can be defined as

$$\begin{bmatrix} M_v + M_1(q) & M_2(q) \\ M_2^T(q) & M_m \end{bmatrix} \begin{bmatrix} \dot{\varsigma} \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} C_v + C_1 & C_2 \\ C_3 & C_m \end{bmatrix} \begin{bmatrix} \varsigma \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} G_v + G_e(q) \\ G_m(q) \end{bmatrix} = \begin{bmatrix} \tau_v \\ \tau_q \end{bmatrix}$$  

(4.36)

or in the more general form

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

(4.37)

where $\dot{\xi} = [\varsigma^T, \dot{\varsigma}^T]^T \in \mathbb{R}^8$ is the acceleration, $\xi = [\varsigma^T, \dot{\varsigma}^T]^T \in \mathbb{R}^8$, $\varsigma = [v^T, \omega^T]^T \in \mathbb{R}^6$, $q \in \mathbb{R}^2$ is the joint position, $\dot{q} \in \mathbb{R}^2$ is the joint velocity, $v$
and \( \omega \in \mathbb{R}^3 \) are the linear and angular velocities, and \( \tau \in \mathbb{R}^8 \) is vector of a force and torque inputs. The system matrices \( M(q), C(q, \dot{q}, \xi), D(q, \dot{q}, \xi), G(q, \zeta) \) are defined below.

Forces and Torques

\( \tau = B_u u \) in (4.37) are the linear forces (thrusts), UAV body torques and joint torques, and \( B_u \) is the actuator input configuration matrix as follows

\[
\tau = \begin{bmatrix} \tau_V \\ \tau_q \end{bmatrix} = \begin{bmatrix} B_V & O_{6 \times 2} \\ O_{2 \times 4} & I_2 \end{bmatrix} \begin{bmatrix} u_V \\ u_q \end{bmatrix} = B_u u \quad \text{and} \quad B = \begin{bmatrix} B_V & O_{6 \times 2} \\ O_{2 \times 4} & I_2 \end{bmatrix}
\] (4.38)

The force and moment acting on the vehicle

\( \tau_V = B_V \cdot u_V \in \mathbb{R}^6, \)

uses \( B_V \in \mathbb{R}^{6 \times 4} \), a special thruster configuration matrix, to distribute the inputs \( u_V \in \mathbb{R}^4 \) which has the following components

\[
u_V = [ u_1, \tau_\phi, \tau_\theta, \tau_\psi ]^T \in \mathbb{R}^4,
\] (4.39)

The vector \( u_q \) is the joint torque,

\[
u_q = [ \tau_{q1}, \tau_{q2} ]^T \in \mathbb{R}^2,
\]

Note that the UAV body is under-actuated in the sense that there are four control inputs and six degrees of freedom.

Inertia Matrix: \( M(q) \)

The inertia matrix is given by

\[
M(q) = \begin{bmatrix} M_v + M_1(q) & M_c(q) \\ M_c^T(q) & M_m(q) \end{bmatrix} \in \mathbb{R}^{8 \times 8}
\] (4.40)

in which the inertia of the vehicle can be defined as

\[
M_v = M_{UV} + M_{VA} \in \mathbb{R}^{6 \times 6}
\] (4.41)

where \( M_{VA} \) is the added inertia due to the loading of the media which can be neglected since the air density about the vehicle is small, and \( M_1(q) \) is the added inertia due to
the manipulator. $M_c \in \mathbb{R}^{6 \times n}$ is the matrix related to the reaction forces and moments between the UAV and manipulator, $M_m \in \mathbb{R}^{n \times n}$ is the inertia matrix of the manipulator. $M_{UV} = \begin{bmatrix} mI_3 & -mS(p_a) \\ mS(p_a) & J_c \end{bmatrix} \in \mathbb{R}^{6 \times 6}$ is the inertia matrix of the UAV, which has the following property

$$M_{UV} = M_{UV}^T > 0 \text{ and } M_{UV} = 0.$$ 

The origin of the quadrotor body-fixed frame is considered to be coincident with the center of mass and then $r_G = O_{3 \times 1}$ which reduces $M_{UV}$ to

$$M_{UV} = \begin{bmatrix} mI_3 & O_3 \\ O_3 & I \end{bmatrix}.$$ 

Coriolis and Centripetal Matrix

The Coriolis and Centripetal matrix is given by

$$C(q, \xi) \in \mathbb{R}^{(6+n) \times (6+n)} = \mathbb{R}^{8 \times 8} \ (n = 2),$$

where

$$C(q, \xi) = \begin{bmatrix} C_v(\zeta) + C_1(q, \dot{q}, \zeta) & C_2(q, \dot{q}) \\ C_3(q, \dot{q}, \zeta) & C_m(q, \dot{q}) \end{bmatrix}, \quad \zeta = \begin{bmatrix} v \\ \omega \end{bmatrix}.$$ 

$C_v(\zeta) \in \mathbb{R}^{6 \times 6}$ is the Coriolis-Centripetal matrix of the UAV body. $C_1(q, \dot{q}, \zeta) \in \mathbb{R}^{6 \times 6}$ is the Coriolis-Centripetal matrix due to the interaction of the UAV and the robot manipulator. $C_2(q, \dot{q}) \in \mathbb{R}^{6 \times 2}$ is the effect of the manipulator on the UAV body, $C_3(q, \dot{q}, \zeta) \in \mathbb{R}^{2 \times 6}$ is the effect of the UAV body on the manipulator, $C_m(q, \dot{q}) \in \mathbb{R}^{2 \times 2}$ Coriolis-Centripetal terms of the robot manipulator.

Damping Effects: $D(q, \xi)$

$$D(q, \xi) = \begin{bmatrix} D_v(\zeta) + D_1(q) + D_2(q, \dot{q}, \zeta) & D_3(q, \dot{q}, \zeta) \\ D_4(q, \dot{q}, \zeta) & D_m(q) + D_5(q, \dot{q}, \zeta) \end{bmatrix}$$

where $D_v(\zeta)$ is the aerodynamic damping and drag matrix. $D_1(q)$ is the damping and drag term due to the configuration of the manipulator links. $D_2(q, \dot{q}, \zeta)$ is the damping due to the interaction between body and manipulator. $D_5(q, \dot{q}, \zeta)$ is the quadratic drag term on manipulator due to UAV motion and $D_m(q)$ is the linear aerodynamic
skin friction affecting the manipulator. The effect of aerodynamic forces on the robot manipulator are assumed to be small since the value of $\rho$ is assumed to be very small for air.

Gravity Forces and Moments: $G(R^T_f, q) \in \mathbb{R}^8$

The gravity forces can be defined as follows:

$$G = \begin{bmatrix} G_v \\ G_m \end{bmatrix} \in \mathbb{R}^{6+2}$$

$G_v$ is the UAV gravitational force and moment vector and $G_E$ is the gravity force and moment vector due to the manipulator. $G_m$ is the manipulator gravitational force. $G_v$ is defined by

$$G_v (R^T) = \begin{bmatrix} f_G \\ p_G \times f_G \end{bmatrix}$$

where

$$f_G = mgR^TE_z = R^T \begin{bmatrix} 0 \\ 0 \\ W \end{bmatrix} = R^T \begin{bmatrix} 0 \\ 0 \\ W \end{bmatrix}$$

$W = mg$ is a scalar $f_G = R^F_1 \begin{bmatrix} 0 \\ 0 \\ W \end{bmatrix}$ and $p_G = [x_G \ y_G \ z_G]^T$ is the position of the center of gravity.

$G_m$ can be determined from

$$p_G \times f_G = \begin{bmatrix} x_G \\ y_G \\ z_G \end{bmatrix} \times R^T \begin{bmatrix} 0 \\ 0 \\ W \end{bmatrix} = \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{bmatrix}$$

$$= \begin{bmatrix} x_G \ y_G \ z_G \\ W(-s\theta) \ Wc\theta s\phi \ Wc\theta c\phi \end{bmatrix}$$

$$= \begin{bmatrix} \hat{x} (y_GWc\theta s\phi - z_GWc\theta c\phi) \\ -\hat{y} (x_GWc\theta c\phi - z_GW(-s\theta)) \\ \hat{z} (x_GWc\theta s\phi - y_GW(-s\theta)) \end{bmatrix}$$

For $n \times 1 (n = 2)$ vector for robot manipulator, the gravity forces are defined as

$$G_m = [f_{g_i}, \ p_{i-1}^i \times f_{g_i}]^T \in \mathbb{R}^2 \text{ and } f_{g_i} = [0, \ 0, \ m_i g]^T$$
Due to the assumption that the body-fixed frame is at the center of gravity

\[ G_v = \begin{bmatrix} f_G \\ O_{3\times1} \end{bmatrix} \in \mathbb{R}^6. \]

**Coordinated Tracking Control**

A closed-loop controller is designed based on the tracking error dynamics of the end-effector of the underactuated UAV and robot manipulator system. Full-state feedback is assumed; that is, all the signals such as position and orientation of the UAV and velocities are assumed to be measurable. It is also assumed that the desired trajectories and up to their second order derivatives are all bounded; i.e., \( p_d(t), \dot{p}_d(t), \) and \( \ddot{p}_d(t) \in \mathcal{L}_\infty \) and \( \Theta_d(t), \dot{\Theta}_d(t), \) and \( \ddot{\Theta}_d(t) \in \mathcal{L}_\infty \). The position and orientation tracking errors are combined to create the end-effector tracking error

\[
e_p = \begin{bmatrix} (R_E^I)^T (p_E - p_D) \\ \Theta_E - \Theta_D \end{bmatrix} \equiv \begin{bmatrix} (R_E^I)^T \\ O_3 \end{bmatrix} \begin{bmatrix} O_3 \\ I_3 \end{bmatrix} \begin{bmatrix} p_E - p_D \\ \Theta_E - \Theta_D \end{bmatrix} \in \mathbb{R}^6. \quad (4.42)
\]

Moreover, by defining the vectors \( x_E(t) = \begin{bmatrix} p_E \\ \Theta_E \end{bmatrix} \in \mathbb{R}^6 \) and \( x_D(t) = \begin{bmatrix} p_D \\ \Theta_D \end{bmatrix} \in \mathbb{R}^6 \), (4.42) yields

\[
e_p = \begin{bmatrix} (R_E^I)^T \\ O_3 \end{bmatrix} \begin{bmatrix} O_3 \\ I_3 \end{bmatrix} (x_E - x_D). \quad (4.43)
\]

Taking the time derivative of the \( e_p(t) \) yields

\[
\dot{e}_p \triangleq \begin{bmatrix} -S(\omega_e) \\ O_3 \end{bmatrix} \begin{bmatrix} O_3 \\ O_3 \end{bmatrix} e_p + e_v \quad (4.44)
\]

where the velocity tracking error, \( e_v(t) \in \mathbb{R}^6 \), is defined from (4.44) as

\[
e_v \equiv \begin{bmatrix} (R_E^I)^T \\ O_3 \end{bmatrix} \begin{bmatrix} O_3 \\ I_3 \end{bmatrix} (\dot{x}_E - \dot{x}_D) \text{ where } \dot{x}_E = \begin{bmatrix} \dot{p}_E \\ \dot{\Theta}_E \end{bmatrix}, \dot{x}_D = \begin{bmatrix} \dot{p}_D \\ \dot{\Theta}_D \end{bmatrix}. \quad (4.45)
\]

From the definition of \( e_v \), differentiating it yields as

\[
\dot{e}_v \triangleq \begin{bmatrix} -S(\omega_e) \\ O_3 \end{bmatrix} \begin{bmatrix} O_3 \\ O_3 \end{bmatrix} e_v + \begin{bmatrix} (R_E^I)^T \\ O_3 \end{bmatrix} \begin{bmatrix} O_3 \\ I_3 \end{bmatrix} (\ddot{x}_E - \ddot{x}_D). \quad (4.46)
\]

Let \( V_1 \) be the positive definite Lyapunov candidate function as

\[
V_1 = \frac{1}{2} e_p^T e_p. \quad (4.47)
\]
Taking time derivative of $V_1$ yields

$$
\dot{V}_1 = e_p^T \dot{e}_p
= e_p^T \left[ \begin{array}{cc} -S(\omega_e) & O_3 \\ O_3 & O_3 \end{array} \right] e_p + e_v
$$

(4.48)

where

$$
\dot{e}_p = \left[ \begin{array}{c} \left( \hat{R}_E^l \right)^T O_3 \\ O_3 \\ I_3 \end{array} \right] (x_E - x_D) + \left[ \begin{array}{c} \left( \hat{R}_E^l \right)^T O_3 \\ O_3 \\ I_3 \end{array} \right] (\dot{x}_E - \dot{x}_D). 
$$

(4.49)

The second equation on the right side of (4.49) yields

$$
\left[ \begin{array}{c} \left( \hat{R}_E^l \right)^T O_3 \\ O_3 \\ I_3 \end{array} \right] (x_E - x_D) = \left[ \begin{array}{cc} -S(\omega_e) & O_3 \\ O_3 & O_3 \end{array} \right] \left[ \begin{array}{c} \left( \hat{R}_E^l \right)^T O_3 \\ O_3 \\ I_3 \end{array} \right] (x_E - x_D),
$$

(4.50)

and

$$
\left( \hat{R}_E^l \right)^T = -S(\omega_e) \left( R_E^l \right)^T.
$$

(4.51)

Then, (4.44) is derived as

$$
\dot{e}_p = \left[ \begin{array}{c} -S(\omega_e) \\ O_3 \end{array} \right] e_p + e_v.
$$

(4.52)

Thus, the function yields

$$
\dot{V}_1 = -e_p^T \left[ \begin{array}{cc} S(\omega_e) & O_3 \\ O_3 & O_3 \end{array} \right] e_p + e_p^T e_v,
$$

(4.53)

where the property of skew-symmetric matrix was applied. Define a filtered tracking error, $r_E \in \mathbb{R}^6$, as

$$
r_E = \left[ \begin{array}{c} r_p \\ e_\Theta \end{array} \right] = \left[ \begin{array}{c} \alpha_1 \left( R_E^l \right)^T (p_E - p_D) + (R_E^l)^T (\hat{p}_E - \hat{p}_D) \\ \alpha_2 (\Theta_E - \Theta_D) + (\hat{\Theta}_E - \hat{\Theta}_D) \end{array} \right],
$$

(4.54)

$$
= \left[ \begin{array}{cc} \alpha_1 & O \\ O & \alpha_2 \end{array} \right] \left[ \begin{array}{cc} \left( R_E^l \right)^T O_3 \\ O_3 \\ I_3 \end{array} \right] (x_E - x_D) + \left[ \begin{array}{cc} \left( R_E^l \right)^T O_3 \\ O_3 \\ I_3 \end{array} \right] (\dot{x}_E - \dot{x}_D),
$$

(4.55)

$$
\equiv \alpha e_p + e_v.
$$

(4.56)

where $\alpha$ is a constant gain matrix given
$\alpha \equiv \begin{bmatrix} \alpha_1 & O \\ O & \alpha_2 \end{bmatrix} \in \mathbb{R}^{6 \times 6}$

where $\alpha_{1,2}$ are each constant matrix. Utilizing (4.52), (4.51) yields

$$\dot{V}_1 = e_p^T (r_E - \alpha e_p) = -\alpha e_p^T e_p + e_p^T r_E$$

(4.53)

where

$$e_p^T = (x_E - x_D)^T \begin{bmatrix} (R_E^T)^T O_3 \\ O_3 \end{bmatrix}^T = (x_E - x_D)^T \begin{bmatrix} R_E^T & O_3 \\ O_3 & I_3 \end{bmatrix}.$$

Then,

$$\dot{V}_1 \leq -\lambda_{\text{min}}\{\alpha\} \|e_p\|^2 + e_p^T r_E.$$  

(4.54)

The last term on the right side in (4.53) will be designed to cancel out by the term, $r_E^T e_p$ as a scaler, in $\dot{V}_2(t)$ when designing the closed-loop feedback controller $u(t)$. The time derivatiation of (4.52) yields

$$\dot{r}_E = \alpha \dot{e}_p + \dot{e}_v.$$  

(4.55)

Substituting (4.46) and (4.50) into that (4.55) yields

The second Lyapunov candidate function is selected by

$$V_2 = \frac{1}{2} r_{E}^T r_{E}. $$

(4.56)

Differentiating (4.56) and then substituting (4.50) and (4.46) in that yields

$$\dot{V}_2 = r_{E}^T \dot{r}_E$$

(4.57)

$$= r_{E}^T \left(-\left[ S(\omega_v) O_3 O_3 \right] (\alpha e_p + e_v) + \alpha e_v + \left[ (R_E^T)^T O_3 O_3 \right] \left( \ddot{x}_E - \ddot{x}_D \right) \right)$$

$$= -r_{E}^T \left[ S(\omega_v) O_3 O_3 \right] r_{E} + \alpha r_{E}^T e_v + r_{E}^T \left[ (R_E^T)^T O_3 O_3 \right] \left( \ddot{x}_E - \ddot{x}_D \right)$$

(4.58)

where (4.52) was used.

End-Effector Kinematics
From (4.33)
\[
\dot{x}_E = \begin{bmatrix} \dot{p}_E \\ \dot{\Theta}_E \end{bmatrix}, \\
= \begin{bmatrix} R^I_F(\Theta) O_3 & \bar{J}_v \\ O_3 T^I_F(\Theta) & \bar{J}_v \end{bmatrix} \begin{bmatrix} v \\ \omega \\ \dot{\theta} \end{bmatrix}, \\
= \begin{bmatrix} R^I_F(\Theta)v + O_3 \times \omega + \bar{J}_v \dot{\theta} \\ O_3 + T^I_F(\Theta)\omega + \bar{J}_\omega \dot{\theta} \end{bmatrix}
\] (4.59)

where the position and orientation rate of the end-effector vector, \( \dot{p}_E \) and \( \dot{\Theta}_E \), are defined in (4.29) and (4.32), respectively.

From the kinematics and the dynamic modeling equation, we can design the control input as follows: rewrite (4.33)
\[
\dot{x}_E = D_E \xi.
\]

Taking the time derivative of \( \dot{x}_E \) yields
\[
\ddot{x}_E = \dot{D}_E \xi + D_E \dot{\xi}
\] (4.60)

where
\[
D_E = \begin{bmatrix} R^I_F & O & \bar{J}_v \\ O & T^I_F & \bar{J}_\omega \end{bmatrix},
\]

and
\[
\frac{d}{dt} (D_E) = \begin{bmatrix} S(\omega) R^I_F & O \\ O & \frac{d}{dt} (T^I_F) \end{bmatrix} \begin{bmatrix} S(\omega) R^I_F \bar{J}_v + \frac{d}{dt} (\bar{J}_v) \\ \frac{d}{dt} (T^I_F) \bar{J}_\omega + \frac{d}{dt} (\bar{J}_\omega) \end{bmatrix},
\] (4.61)
in which
\[
\frac{d}{dt} (T^I_F) = \frac{d}{dt} \left( \begin{bmatrix} 1 & s\phi t\theta \\ 0 & c\phi \end{bmatrix} \begin{bmatrix} c\phi & s\phi \\ \frac{s\phi}{c\theta} & \frac{c\phi}{c\theta} \end{bmatrix} \right),
\] (4.62)

and
\[
\frac{d}{dt} (\bar{J}_v) = \frac{d}{dt} \left( \begin{bmatrix} \bar{J}_{v2} \\ \bar{J}_{v3} \end{bmatrix} \right) = \left[ \frac{\partial (z_1 \times (O_3 - O_1))}{\partial q_1}, \frac{\partial (z_2 \times (O_3 - O_1))}{\partial q_2} \right] \dot{q},
\]
\[
\frac{d}{dt} (\bar{J}_\omega) = \frac{d}{dt} \left( \begin{bmatrix} \bar{J}_{\omega2} \\ \bar{J}_{\omega3} \end{bmatrix} \right) = \left[ \frac{\partial z_1}{\partial q_1}, \frac{\partial z_2}{\partial q_2} \right] \dot{q}
\] (4.63)

where \( \bar{J}_v, \bar{J}_\omega \) are given from \( J_q(t) \) in (4.1) for the robot joint velocities. Note that the time derivative of Jacobian matrix can use the second and third column vectors of
related to the robot manipulator. \( \dot{\xi}(t) \) is obtained from the integrated modeling equation in (4.37) as a single matrix form

\[
\dot{\xi} = M^{-1}(\tau - (C + D)\xi - G).
\]

(4.64)

Substitute (4.64) into (4.60) then substitute (4.34) for \( \xi \) in \( \ddot{x}_E \)

\[
\ddot{x}_E = \dot{D}_E \xi + D_E M^{-1} [\tau - (C + D)\xi - G].
\]

(4.65)

Substituting the right side equations into the (4.58) for \( \ddot{x}_E(t) \) yields

\[
\dot{V}_2 = \alpha r_E^T e_v + r_E^T \left[ \begin{array}{c}
R_E^T O_3 \\
O_3 \\
I_3
\end{array} \right] \left( \dot{D}_E D_E^\dagger \dot{x}_E + D_E M^{-1} \left[ \tau - (C + D)D_E^\dagger \dot{x}_E - G \right] - \ddot{x}_D \right)
\]

(4.66)

where \( R_E^T = (R_E^l)^T \) and a skew-symmetric property made the first term zero in (4.58).

Design of \( \tau(t) \) Based on Lyapunov Approach

Design of \( \tau(t) \) from (4.58)

\[
\tau = \left[ (C + D)D_E^\dagger \dot{x}_E + G \right] + MD_E \left\{ \ddot{x}_D - \dot{D}_E D_E^\dagger \dot{x}_E - \left[ \begin{array}{c}
R_E^l O_3 \\
O_3 \\
I_3
\end{array} \right] (r_E + e_p + \alpha e_v) \right\}
\]

(4.67)

where

\[
\dot{D}_E = \begin{bmatrix}
S(\omega)R_E^l & O \\
O & \frac{d}{dt} (T_F^l)
\end{bmatrix}
\]

\[
= \begin{bmatrix}
S(\omega)R_E^l J_V + R_E^l \dot{J}_V \\
\frac{d}{dt} (T_F^l) J_\omega + T_F^l \dot{J}_\omega
\end{bmatrix},
\]

and from (4.38) the control input is

\[
u = B^\dagger \tau.
\]

(4.68)

where \( B^\dagger = (B^T B)^{-1} B^T \in \mathbb{R}^{8 \times 6} \). Substituting the right equations of (4.67) into 4.66 and arranging yields

\[
\dot{V}_2 = -r_E^T M r_E - r_E^T e_p,
\]

(4.69)

\[
\leq -\lambda_{\min}\{M\} \|r_E\|^2 - r_E^T e_p
\]
where $\lambda_{\text{min}}\{M\}$ is the minimum value of eigenvalue of inertia matrix, $M$. Combining the two Lyapunov candidate functions, $V_1$ with $V_2$, yields

$$V = V_1 + V_2 = \frac{1}{2}(e_p^T e_p + r_E^T r_E),$$  \hspace{1cm} (4.70)

and then summing up (4.54) and (4.69) yields

$$\dot{V} = e_p^T \dot{e}_p + r_E^T \dot{r}_E;$$

$$\leq -\lambda_{\text{min}}\{\alpha\} \|e_p\|^2 - \lambda_4 \|r_E\|^2. \hspace{1cm} (4.71)$$

Note that $r_E^T e_p$ is cancelled out as a scalar. Let $z = [e_p^T, r_E^T]^T \in \mathbb{R}^{12}$ and $\lambda_5 = \min \{\lambda_{\text{min}}\{\alpha\}, \lambda_4\}$, then, $\dot{V}(t)$ yields

$$\dot{V} \leq -\lambda_5 (\|e_p\|^2 + \|r_E\|^2),$$

$$\leq -\lambda_5 \|z\|^2. \hspace{1cm} (4.72)$$

Using Barbara’s Lemma [8] with (4.72) and (4.70), the closed-loop control input, $\tau(t)$, ensures that the tracking error $z(t)$ goes asymptotically stable: $\|z\| \to 0$ as $t \to \infty$.

According to $z(t)$ is bounded, its member $e_p$ and $r_E$ are bounded owing to $R$ is full rank and $T^{-1}(\Theta)$ exists, which leads $x_E$, $\Theta_E$ and $e_v \in \mathcal{L}_\infty$ due to $x_D \in \mathcal{L}_\infty$ due to the assumption that all desired trajectories are bounded. $\dot{x}_E$ and $\dot{e}_p(t) \in \mathcal{L}_\infty$ due to $e_v \in \mathcal{L}_\infty$. $v, \omega, \dot{q}$ are bounded also $D_E, \xi \in \mathcal{L}_\infty$ due to $\dot{x}_E$ is bounded. $\omega, v_E$ are bounded due to $\dot{e}_p(t), e_p, e_v \in \mathcal{L}_\infty$. Hence, all tracking errors are bounded.

**Conclusion**

This chapter described the development of the unmanned aerial-robotic system. A model of the combined UAV and robot manipulator is proposed from which a coordinated controller of the integrated nonlinear system is developed using a Lyapunov-type method. The design goal for this controller is to simultaneously control the two degree-of-freedom robot manipulator (RM) and the quadrotor Unmanned Aerial Vehicle (UAV) to create a six degree-of-freedom UAV-Robot Manipulator (UAVRM). If
the integration of robot manipulator and aerial robot system is possible, the system could be fast moving and avoid obstacles, but also useful for manipulating physical systems once in place, e.g., changing a lightbulb on a radio tower. This is the ultimate purpose of the work. The UAVRM end-effector can track three desired positions and three angles using feedback signals.


Future Work

Simulations to verify the modeling of the UAVRM system to validate the mathematical modeling presented in this dissertation have not been conducted. This will be an important next step. Also the implementation of the controller for tracking in various conditions should be tested. Finally, force control of the UAVRM end-effector would add to the robustness and usefulness of the proposed system.

Conclusion

In this dissertation, an integrated nonlinear control system for an unmanned aerial vehicle (UAV) and robot manipulator is proposed. The dynamic model of the integrated UAVRM system is derived and a tracking controller for the end-effector was demonstrated mathematically based on Lyapunov-type approach. The mathematical result shows that a GUUB tracking result is achieved. After modeling of each subsystem, we integrate both mechanisms into a single dynamic, coordinated model. The design goal for this controller was to add a two degree-of-freedom robot manipulator (RM) to the quadrotor Unmanned Aerial Vehicle (UAV) to create a single UAV-Robot Manipulator (UAVRM) system. The UAVRM end-effector can track three desired positions and three angles. The UAV is equipped with robotic arms that can be used for applications where it is dangerous or inconvenient to use direct human intervention; e.g., changing a lightbulb on a radio tower would be a good example. A growing suite of applications can be envisioned in both military and civilian applications. To our best knowledge, this is a first paper to present such a comprehensive modeling and control design for a quadrotor-type helicopter UAV system.

Notation and Nomenclature

\( F \)  
UAV Body-fixed frame

\( G \)  
UAV body-fixed frame

\( I \)  
Earth-fixed (Inertial) frame

\( p_i, \Theta_i \)  
Position and orientation \((i\) is the location or frame\)

\( \dot{p}_i, \dot{\Theta}_i \)  
Position and orientation rate

\( v^i, \omega^i \)  
Velocities denoting airborne quantities using superscript

\( v^i_{jk}, \omega^i_{jk} \)  
Velocities expressed in \(i\) denoting quantities between \(j\) and \(k\)

\( R_{i}^{i-1} \)  
Rotation matrix from the origin \(i\) to the origin \(i - 1\) coordinate frame

\( T_{FI}^{I} \)  
Orientation matrix from the origin \(F\) to the origin \(I\) frame

\( x_i \)  
Position and Orientation of the \(i\) coordinate frame expressed in Earth-fixed frame
APPENDICES
Appendix A

Rigid-body Equation of Motion

Modeling of Quadrotor-type Helicopter UAV

In this appendix, a general dynamic model of the rigid-body UAV is proposed. Based on this general UAV rigid-body model, the mechanical characteristics specific to a quadrotor-type UAV are used to derive the quadrotor dynamic model. To derive the modeling equation and mechanics, standard modeling literature ([1], [2], [3], [7]) was used to develop the equations of motion for the quadrotor-type helicopter UAV to fit the control of the integrated UAVRM system. Since the UAV as well as robot manipulator is not fixed in an earth-fixed reference frame, the modeling equations expressed in the body-fixed frame are useful and convenient to express the quantities such as acceleration, velocities, torques. In addition, in order to obtain the equation of motion of the UAVRM, a matrix form of the the equation of motion is expressed in the body-fixed frame for computational easy.

Rigid-body Dynamics

Figure A.1 shows the Earth-Fixed \((x_I, y_I, z_I)\) coordinate frame labeled as "\(I\)", the Body-Fixed coordinate frame \((x_b, y_b, z_b)\) labeled as "\(B\)”, and the Center of Gravity \((x_G, y_G, z_G)\) coordinate frame labeled as "\(G\)”. The vectors \(v\) and \(\omega\) are the linear and angular velocity in \(B\), \(v_C\) is the linear velocity in \(G\), \(p\) is the position of an arbitrary particle in \(B\), \(p_b\) is the location of the origin of \(B\) in \(I\), and \(p_G\) is the origin of \(G\) in \(B\), and \(p_C\) is the origin of \(G\) in \(I\). The assumptions made for the modeling are as follows:

1. The vehicle is a uniformly dense rigid body and the mass is constant. Thus, the distance, \(p_G\), from the origin \(B\) of the body fixed frame to the center of gravity of the vehicle can be expressed in the inertial frame, defined as

\[
p^I_G = \frac{1}{m} \int_{V_b} p^I \cdot \rho_A dV \quad (A.1)
\]

where \(m\) is the mass of the rigid-body vehicle, \(p^I\) is the position expressed in earth-fixed frame \((I)\), \(\rho_A\) is the density of the rigid-body, and \(V_b\) is the volume...
Figure A.1 Rigid-body Coordinate Frames: Earth-Fixed Inertial, Body-fixed, and Center of Gravity
of the body in which \( m \) is the total mass defined as

\[
m = \int_{V_b} \rho_A dV \tag{A.2}
\]

2. The earth-fixed frame is considered the same as the inertial frame.

Equations of motion are derived in the body-fixed frame to obtain the advantage of certain geometric properties. The following equation of motion is expressed in the body-fixed frame based on Newton’s Second Law and Euler’s axioms, respectively as

\[
m \dot{v}^l_C = f_C
\]

and

\[
J_C \dot{\omega}^l_C = m_C
\]

where \( f_C \in \mathbb{R}^3 \) is the external force acting on the center of gravity (COG) (denoted with the subscript “\( C \)”), and \( m_C \in \mathbb{R}^3 \) is the vector of moments acting on the body referred to the body center of gravity given in body-fixed frame, \( \dot{v}^l_C \) is the acceleration in the earth-fixed frame \((I)\), \( \dot{\omega}^l_C \) is the angular acceleration with respect to the frame \( I \), and \( J_C \) is the inertia tensor referenced to the body center of gravity.

Rigid-Body Translational Dynamics

From Figure A.1, the position vectors expressed in the inertial frame \((I)\) are given by

\[
p^l_C = p^l_b + p^l_G.
\]

By taking the time derivative, the velocity of the center of gravity is

\[
\dot{p}^l_C = \dot{p}^l_b + \dot{p}^l_G.
\]

Using the facts

\[
\dot{p}^l_C = v^l_C \text{ and } \dot{p}^l_b = v^l_b,
\]

the velocity of the center of gravity can be written as

\[
v^l_C = v^l_b + \omega \times p_G
\]
where
\[ \dot{p}_G^l = \dot{p}_G + \omega \times p_G = \omega \times p_G \tag{A.9} \]

Equation A.9 shows a useful form for the derivative of \( P_G^l \) that will be utilized later. \( \dot{p}_G = 0 \) due to rigid-body expressed in body-frame. Then, the velocity of the vehicle is
\[ v = \dot{p}_b^l + \dot{p}^l = v_b^l + \omega \times p \tag{A.10} \]

where \( \dot{p} = 0 \). Differentiating \( v_C^l (t) \) yields
\[ \dot{v}_C^l = \dot{v}_b^l + \omega \times p_G + \omega \times \dot{p}_C^l = (\dot{v}_b + \omega \times v_b) + \dot{\omega} \times p_G + \omega \times (\omega \times p_G) \tag{A.11} \]

where the vector property shown in A.9 was used for \( \dot{v}_b^l \) and \( \dot{p}_G^l \). Substituting \( \dot{v}_C^l \) into (A.3) in Newton-Euler formulation yields the matrix form for the translational dynamics
\[ m [ \dot{v}_b + S(\omega)v_b - S(p_G)\dot{\omega} + S(\omega)S(\omega)p_G ] = f_C \]
\[ m \ddot{v}_b - mS(p_G)\ddot{\omega} + mS(\omega)v_b - mS(\omega)S(p_G)\omega = f_C \]

where the following vector properties were used as
\[ \omega \times (\omega \times p_G) = S(\omega)S(\omega)p_G = -S(\omega)S(p_G)\omega \]

in which
\[ a \times b = S(a)b = -S(b)a \text{ where } a, b \text{ are vectors.} \]

Hence, arranging the modeling equation yields
\[ \begin{bmatrix} mI_{3 \times 3}, & -mS(p_G) \end{bmatrix} \begin{bmatrix} \dot{v}_b \\ \dot{\omega} \end{bmatrix} + \begin{bmatrix} mS(\omega), & -mS(\omega)S(p_G) \end{bmatrix} \begin{bmatrix} v_b \\ \omega \end{bmatrix} = f_C. \tag{A.12} \]

where \( I_{n \times n} \) is the \( n \times n \) identity matrix.
Quadrotor Helicopter Coordinate Frames

Quadrotor Helicopter Translational Dynamics

Owing to the symmetrical structure of the quadrotor-type helicopter, the origin of the body fixed frame with the center of gravity can be assumed to coincide; \( B = G \) as shown in Figure A.2. Then, the modeling equation is simplified as

\[
\begin{align*}
    p_G &= [0, 0, 0]^T, \\
    v_b &= v_C, \\
    p_b &= p_C, \\
    \dot{p}_b^I &= \dot{p}_C^I, \text{ and } v_b^I &= v_C^I.
\end{align*}
\]

Applying this to the \( m\dot{v}_C^I \) yields

\[
m(\dot{v}_C + \omega \times v_C) = f_C. \tag{A.13}
\]

Using \( N_1(v, |v|, \theta_i) \) to encompass the nonlinear aerodynamic forces where \( \theta_i \) is the constant damping parameter in the translational dynamics (specific models are used in Chapter 2 and Chapter 3) and \( G_1 \) to represent gravity forces, substituting into (A.13) yields the dynamic modeling equation of the quadrotor expressed in the body-fixed frame as

\[
m(\dot{v}_C + \omega \times v_C) + N_1(v, |v|, \theta_i) - G_1 = f_C \tag{A.14}
\]
Arranging (A.14) and using the matrix $S(\omega) \in \mathbb{R}^{3 \times 3}$ in lieu of the cross product yields

$$m \dot{v}_C = -mS(\omega)v_C - N_1(v, |v|, \theta_i) + G_1 + f_C,$$

(A.15)

from which the three components are written as

$$m \dot{v}_1 = -m(v_2 \omega_3 - v_3 \omega_2) - N_1(v_1, |v|, \theta_1) - mg \sin(\theta) = 0$$

$$m \dot{v}_2 = -m(v_3 \omega_1 - v_1 \omega_3) - N_1(v_2, |v|, \theta_2) + mg \cos(\theta) \sin(\phi) = 0$$

$$m \dot{v}_3 = -m(v_1 \omega_2 - v_2 \omega_1) - N_1(v_3, |v|, \theta_3) + mg \cos(\theta) \cos(\phi) + u_1 = 0$$

Note that the gravity forces $G_1(R)$ were expanded utilizing the Directional Cosine Matrix $R^T$ which was given in (2.2).

Rigid-Body Rotational Dynamics

Using Figure A.1, the rotational motion can be derived in a manner similar to the translational dynamics.

**Rotation about a fixed point**

At the point $p$, the angular momentum of the rigid body about the origin $B$ is the integral of the angular momenta

$$H_b = \int_v p \times v \rho_A dv$$

(A.17)

where $p$ is the particle’s position relative to its reference frame $B$, $v$ is the velocity and $\omega$ is the rigid body’s angular velocity about the point. Differentiating (A.17) to find the rate of change of angular momentum about $B$ yields

$$M_B = \frac{d}{dt}(H_B), v = \dot{v}_b + \dot{p}_b + \omega \times p$$

$$= \int_v \dot{p}_v \times v \rho_A dv + \int_v p \times \dot{v} \rho_A dv$$

$$= m_b + \int_v \dot{p}_v \times v \rho_A dv$$

(A.18)

where the second equation on the right side in the second row is the moment $m_b$ referred to the body-fixed frame. After substituting the velocity of (A.10) in the translational dynamics, the moment $M_B$ yields

$$M_B = m_b + \int_v \dot{p}_v \times (\dot{p}_b^I + \dot{p}_I) \rho_A dv$$

$$= m_b - \dot{v}_b^I \int_v \dot{p}_v^I \rho_A dv + \int_v \dot{p}_v^I \times \dot{p}_v^I \rho_A dv$$

$$= m_b - \dot{v}_b^I \times \int_v \dot{p}_v^I \rho_A dv$$
in which the cross vector property, \( \dot{p}' \times \dot{p}' = O_{3 \times 1} \), was used and \( \dot{p}'_b = v'_b \). From the definition \( p'_G \) in Assumption 2, rewrite the moment of mass as

\[
mp'_G = \int_V p' \rho_A dv. \tag{A.19}
\]

Taking the time derivative of the position vector produces

\[
m \dot{p}'_G = \int_V \dot{p}' \rho_A dv \tag{A.20}
\]

Thus, utilizing the definition in (A.9), (A.20) yields

\[
m(\omega \times p_G) = \int_V \dot{p}' \rho_A dv \tag{A.21}
\]

After substituting (A.21) for \( \int_V \dot{p}' \rho_A dv \) in \( M_B \), the moment yields

\[
M_B = m_b - mv'_b \times (\omega \times p_G) \tag{A.22}
\]

The absolute momentum \( H_b \) in (A.17) can be expressed by using the definition (A.10) in the translational dynamics as

\[
H_b = \int_V p \times (v'_b + \omega \times p) \rho_A dv
= \int_V (p \times v'_b) \cdot \rho_A dv + \int_V p \times (\omega \times p) \rho_A dv
= \int_V p \rho_A dv \times v'_b + J_C \omega
= mp'_G \times v'_b + J_C \omega
\]

where the last equation on the right side of the second row is defined by [3] as

\[
J_C \omega = \int_V p \times (\omega \times p) \rho_A dv
\]

in which \( J_C \) is assumed to be constant with respect to time. Differentiating \( H_b \) produces

\[
\dot{H}_b = mp'_G \times v'_b + mp'_G \times \dot{v}'_b + J_C \dot{\omega}'
= m (\omega \times p_G) \times v'_b + \dot{m} p_G \times (\dot{v}_b + \omega \times v_b) + [J_C \dot{\omega} + \omega \times (J_C \omega)]. \tag{A.23}
\]
Rewriting the moment using the vector property \( m(\omega \times p_G) \times v'_b = -mv'_b \times (\omega \times p_G) \) yields
\[
M_B = J_C \dot{\omega} + \omega \times (J_C \omega) - mv'_b \times (\omega \times p_G) + mp_G \times (\dot{v}_b + \omega \times v_b).
\] (A.24)

By substituting (A.22) in (A.24) to remove \( M_B \) and hence canceling out \( mv'_b \times (\omega \times p_G) \) yields
\[
J_C \dot{\omega} + \omega \times (J_C \omega) + mp_G \times (\dot{v}_b + \omega \times v_b) = m_b.
\] (A.25)

After rearranging the terms, the rotational dynamics in (A.25) becomes
\[
mS(p_G)\dot{v}_b + J_C \dot{\omega} + mS(p_G)S(\omega)v_b - S(J_C) \omega = m_b
\] (A.26)

where the vector property A.9 was used. Thus, the matrix form for rotational dynamics is
\[
\begin{bmatrix}
  mS(p_G), & J_C
\end{bmatrix}
\begin{bmatrix}
  \dot{v}_b \\
  \dot{\omega}
\end{bmatrix}
+ \begin{bmatrix}
mS(p_G)S(\omega), & -S(J_C) \omega
\end{bmatrix}
\begin{bmatrix}
v_b \\
\omega
\end{bmatrix}
= m_b
\] (A.27)

Quadrotor Helicopter Rotational Dynamics

In a similar fashion in the translational dynamics, the rotational dynamic model of the quadrotor-type helicopter is given by
\[
J_C \dot{\omega} = S(J_C) \omega + m_C.
\] (A.28)

When considering the aerodynamic forces, \( N_2(v, |v|, \theta_j) \), as a specified model of aerodynamic forces, (A.28) yields
\[
J_C \dot{\omega} = S(J_C) \omega - N_2(v, |v|, \theta_j) + m_C
\] (A.29)

where \( \theta_j \) is the constant damping parameter in the rotational dynamics. In addition, the inertia matrix is simplified as
\[
J_C = \begin{bmatrix}
  J_1 & 0 & 0 \\
  0 & J_2 & 0 \\
  0 & 0 & J_3
\end{bmatrix}
\] (A.30)
by assuming that the off-diagonal terms are very smaller that the principal term due to symmetrical structure of the quadrotor. Note that the gravity terms are a zero vector. Hence,

\[
J_1 \dot{\omega}_1 = -(J_3 - J_2)\omega_2 \omega_3 - N_2(v_2, |v|, \theta_1) + \tau_1 \\
J_2 \dot{\omega}_2 = -(J_1 - J_3)\omega_1 \omega_3 - N_2(-v_1, |v|, \theta_2) + \tau_2 \\
J_3 \dot{\omega}_3 = -(J_2 - J_1)\omega_1 \omega_2 + \tau_3
\] (A.31)

Combined Translational with Rotational Rigid-Body Dynamics

The matrix form of the combined rigid-body UAV dynamics as a six-degrees-of-freedom nonlinear dynamic equation of motion can be modeled by gathering (A.12) and (A.27) as

\[
\begin{bmatrix}
  mI_{3\times 3} & -mS(p_G) \\
  mS(p_G) & J_C
\end{bmatrix}
\begin{bmatrix}
  \dot{v}_b \\
  \dot{\omega}
\end{bmatrix}
= \begin{bmatrix}
  -mS(\omega) & mS(\omega)S(p_G) \\
  -mS(p_G)S(\omega) & S(J_C\omega)
\end{bmatrix}
\begin{bmatrix}
  v_b \\
  \omega
\end{bmatrix}
+ \begin{bmatrix}
  f_C \\
  m_b
\end{bmatrix}
\] (A.32)

and the combined quadrotor UAV translational and rotational dynamics with aerodynamic and gravity forces and moments is given by

\[
\begin{bmatrix}
  mI_{3\times 3} & O_3 \\
  O_3 & J_C
\end{bmatrix}
\begin{bmatrix}
  \dot{v}_b \\
  \dot{\omega}
\end{bmatrix}
= \begin{bmatrix}
  -mS(\omega) & O_3 \\
  O_3 & S(J_C\omega)
\end{bmatrix}
\begin{bmatrix}
  v_b \\
  \omega
\end{bmatrix}
- \begin{bmatrix}
  N_1 \\
  N_2
\end{bmatrix}
- \begin{bmatrix}
  G_1 \\
  O_{3\times 1}
\end{bmatrix}
+ \begin{bmatrix}
  u_1 \\
  u_2
\end{bmatrix}
\] (A.33)

where \(I_{3\times 3}\) is the 3 \(\times\) 3 identity matrix, \(O_n\) is \(n \times n\) zero matrix, and \(O_{n1\times n2}\) is the \(n1 \times n2\) zero vector. The simplified equation is

\[
M\dot{\zeta} + C(\omega)\zeta + N(v, |v|, \theta_{ij})\zeta + G = \tau
\] (A.34)

where \(\zeta \in \mathbb{R}^6\) is the velocity defined as

\[
\zeta = \begin{bmatrix}
  v^T \\
  \omega^T
\end{bmatrix}^T
\]

and

\[
v = \begin{bmatrix}
  v_1 \\
  v_2 \\
  v_3
\end{bmatrix}^T \in \mathbb{R}^3 \\
\omega = \begin{bmatrix}
  \omega_1 \\
  \omega_2 \\
  \omega_3
\end{bmatrix}^T \in \mathbb{R}^3.
\]
Also,

\[ M = \begin{bmatrix} mI_{3 \times 3} & O_3 \\ O_3 & J_C \end{bmatrix}, \quad C(\omega) = \begin{bmatrix} mS(\omega) & O_3 \\ O_3 & -S(J_C \omega) \end{bmatrix}, \quad N = \begin{bmatrix} N_1 \\ N_2 \end{bmatrix}, \quad G = \begin{bmatrix} G_1 \\ O_{3 \times 1} \end{bmatrix}, \]

and \( \tau = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \)

where the following properties of the system matrix can be defined as

\[ M_v = M_v^T > 0 \text{ and } \dot{M} = 0 \]
\[ C_\zeta = -C_v^T(\zeta) \]

Some useful properties of vector used in the above derivations are listed below.

1. If \( \dot{p}^I \) is an arbitrary vector \( p \), defined in an earth-fixed frame \( (I) \), and \( \dot{p} \) is the body-fixed moving frame, then

\[ \dot{p}^I = \dot{p} + \omega \times p \]

2. In a similar manner,

\[ \dot{\omega}^I = \dot{\omega} \]

from \( \omega \times \omega = 0 \)

3. \( S(a) \cdot b = a \times b = -b \times a \) (a, b vectors)
Appendix B

Equations of Motion of Two-Link Robot Manipulator

In this section, the equations of motion for a two-link manipulator are derived. Consider the two-link articulated robot manipulator shown in the Figure B.1 where \( q_1, q_2 \) are the joint angles of the first and second links, \( m_1 \) and \( m_2 \) are the masses of each link, \( g \) is the gravitational acceleration, \( l_i \) is the distance between link origin to center of mass of link \( i \), \( a_1 \) and \( a_2 \) are the link lengths, and \( \tau_1 \) and \( \tau_2 \) are the torques that can be applied by the actuators to each link. The following assumptions are made regarding the dynamic model

1. Each joint type is revolute,
2. Mass of each link is located at \( l_1, l_2 \) and not at the distal end of the joint.

Euler-Lagrange Equations of Motion

The following equation is used for developing the dynamic model of a robot manipulator system [9]

\[
\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = \tau_i; \quad i = 1, \ldots, k
\]

where \( q_i \) is the joint angle, \( \tau_i \) is the torque associated with \( q_i \), and \( i \) is the joint index ordered and increasing from the base to the end-effector. The Lagrangian, \( L \), is defined as

\[
L = K - P_m
\]

in which \( K \) is kinetic energy and \( P_m \) is the potential energy.

Moment of Inertia of Robot Manipulator

\( I \) is the moment of inertia expressed in the body-fixed frame. It is a constant matrix, independent of motion of the object, with the following components:

\[
I = \begin{bmatrix}
I_x & I_{xy} & I_{xz} \\
I_{yx} & I_y & I_{yz} \\
I_{zx} & I_{zy} & I_z
\end{bmatrix}
\] (B.1)
Figure B.1 2-link Robot Manipulator
where $I_x, I_y,$ and $I_z$ are the principal moments of inertia and $I_{xy} = I_{yx}$, etc. are the cross products. The principal moments of inertia for the manipulator are given by

$$I_x = I_y = \frac{1}{4} m_i r_i^2 + \frac{1}{12} m_i h_i^2, \quad I_z = \frac{1}{2} m_i r_i^2. \quad (B.2)$$

The moment of inertia is simplified by symmetry considerations [9] which show the cross terms to be zero, the moment of inertia for link $i$ is

$$I_i = \begin{bmatrix} \frac{1}{4} m_i r_i^2 + \frac{1}{12} m_i h_i^2 & 0 & 0 \\ 0 & \frac{1}{4} m_i r_i^2 + \frac{1}{12} m_i h_i^2 & 0 \\ 0 & 0 & \frac{1}{2} m_i r_i^2 \end{bmatrix}. \quad (B.3)$$

Two Link Robot Manipulator

The Lagrangian equation is based on total energy which consists of kinetic energy and potential energy. The kinetic energy is based on $v_i$ and $\omega_i$, the linear and angular velocities. These velocities are expressed in terms of the Jacobian matrix: $v_i = J_{v_i}(q)\dot{q}, \quad \omega_i = J_{\omega_i}\dot{q}$. Thus, the kinetic energy can be defined as

$$K = \frac{1}{2} \dot{q}^T \left( \sum_{i=1}^{2} (m_i J_{v_i}^T J_{v_i} + J_{\omega_i}^T R_i I_i R_i^T J_{\omega_i}) \right) \dot{q} = \frac{1}{2} \dot{q}^T M_m(q) \dot{q} \quad (B.4)$$

where $I_i$ is the inertia of the link $i$ and $R_i$ is the transformation from the body-fixed frame $i$ to the inertia frame. The properties of the inertia matrix, $M_m(q)$, from the above equation, is symmetric and positive definite with kinetic energy $K > 0$. The potential energy is defined as

$$P_m = \sum_{i=1}^{2} P_i$$

where it is assumed that the mass of the object is concentrated at its center of mass.

Kinematics

First Link Transformation Matrix

To develop the kinematics, the Denavit-Hartenberg convention is used as shown in Table B.1. From the values of Table B.1 for the first link, a rotation, a translation, and
another rotation are required. In general, The following rotation matrix represents a rotation by angle $\theta$ about $z$-axis:

$$R_{z,\theta} = \begin{bmatrix} C\theta & -S\theta & 0 & 0 \\ S\theta & C\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where $C \cdot = \cos(\cdot)$ and $S \cdot = \sin(\cdot)$ were used.

In the first rotation, the angle of rotation is $\theta = q_1$ about $z$-axis and the angle from $x_0$ frame to $x_1$ frame about $z_0$-axis was rotated by $90^\circ$ in CCW direction as seen along $z_0$. Thus, the first rotation is defined as

$$R_{z,\theta=q_1+90^\circ} = \begin{bmatrix} C(q_1 + 90^\circ) & -S(q_1 + 90^\circ) & 0 & 0 \\ S(q_1 + 90^\circ) & C(q_1 + 90^\circ) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -S_{q_1} & -C_{q_1} & 0 & 0 \\ C_{q_1} & -S_{q_1} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(B.5)

Second, the translation by a distance $a_1$ along the $x$-axis is performed and the translational matrix is defined as

$$T_{x,a_1} = \begin{bmatrix} I_{3\times3} \\ O_{1\times3} \end{bmatrix} \begin{bmatrix} a_1 & 0 & 0 \\ 1 & \end{bmatrix}^T$$

(B.6)

where $I_{3\times3}$ is the $3 \times 3$ identity matrix and $O_{1\times3}$ is the $1 \times 3$ zero vector. Finally rotating $90^\circ$ in the CCW direction about $x$-axis from $z_0$ to $z_1$, i.e. $\alpha = -90^\circ$ in the rotation matrix, yields:

$$R_{x,\alpha=-90^\circ} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C\alpha & -S\alpha & 0 \\ 0 & S\alpha & C\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$  

(B.7)

Thus, multiplying the homogeneous equations for the first link from (B.5), (B.6), and (B.7) produces

<table>
<thead>
<tr>
<th>Link $(i)$</th>
<th>$\theta_{zi,i-1}^i$ (angle)</th>
<th>$d$ (offset)</th>
<th>$a/b$ (length)</th>
<th>$\alpha_{zi,i-1}^i$ (twist)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$q_1+90^\circ$</td>
<td>0</td>
<td>$a_1$</td>
<td>$+90^\circ$</td>
</tr>
<tr>
<td>2</td>
<td>$q_2$</td>
<td>0</td>
<td>$b_2$</td>
<td>$-90^\circ$</td>
</tr>
</tbody>
</table>

Table B.1 Denavit-Hartenberg Table for 2-link Robot Manipulator Kinematics
\[
T_1^0 = A_1 = R_{z, \theta} \cdot T_{x,a1} \cdot R_{x,\alpha=+90^\circ} = \begin{bmatrix}
-Sq_1 & 0 & Cq_1 & -Sq_1 \cdot a_1 \\
Cq_1 & 0 & -Sq_1 & Cq_1 \cdot a_1 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\] (B.8)

where the link rotation matrix, \( R_1^0 \), of the first link is given as
\[
R_1^0 = \begin{bmatrix}
-Sq_1 & 0 & Cq_1 \\
Cq_1 & 0 & -Sq_1 \\
0 & 1 & 0
\end{bmatrix}.
\] (B.9)

From (B.8), we can find the axis of rotation, \( z_1 \), and the origin of the first link, \( o_1 \) as
\[
z_1 = \begin{bmatrix}
Cq_1 \\
Sq_1 \\
0
\end{bmatrix}, \\
o_1 = \begin{bmatrix}
-Sq_1 \cdot a_1 \\
Cq_1 \cdot a_1 \\
0
\end{bmatrix}.
\] (B.10)

Note that from the given Figure B.1, \( o_0 \) is the base reference frame and hence
\[
z_0 = \begin{bmatrix}
0 \\
0 \\
1
\end{bmatrix}, \\
o_0 = \begin{bmatrix}
0 \\
0 \\
0
\end{bmatrix}
\] (B.11)

where \( z_0 \) is the axis of rotation according to the basis coordinate, \( o_0x_0y_0z_0 \), and \( o_0 \) is the origin of the frame. Note that the origin of the manipulator will be mapped to the position of the UAV when it is attached to the rigid-body UAV.

**Second Link Translation Matrix**

Based on the D-H values for the second link, first the angle of rotation about \( z \)-axis is \( \theta = q_2 \) and the rotation angle from \( x_1 \)-axis to \( x_2 \)-axis is \( 0^\circ \) as seen from \( z_1 \).

Thus, the first rotation matrix is defined as
\[
R_{z,\theta=q_2+0^\circ} = \begin{bmatrix}
Cq_2 & -Sq_2 & 0 & 0 \\
Sq_2 & Cq_2 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}.
\] (B.12)

The pure translation \( b_2 \) along \( z \)-axis is defined as
\[
T_{x,b2} = \begin{bmatrix}
I_{3\times3} \\
O_{1\times3}
\end{bmatrix} \begin{bmatrix}
0 \\
b_2 \\
1
\end{bmatrix} = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & b_2 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}.
\] (B.13)
The rotation about $x$-axis rotating $90^\circ$ CCW direction yields $\alpha_2 = -90^\circ$, thus the second rotation matrix is

$$R_{x,\alpha_2 = 90^\circ} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (B.14)$$

Thus, the second homogeneous transformation matrix $A_2$ is defined as

$$A_2 = R_{z,\theta_2}T_{z,d_2}R_{x,\alpha_2 = 90^\circ} = \begin{bmatrix} Cq_2 & 0 & Cq_1 & b_2Cq_2 \\ Sq_2 & 0 & Sq_1 & b_2Sq_2 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (B.15)$$

Hence, using the matrices from (B.12), (B.13), and (B.14), the transformation matrix that expresses position and orientation of end effector frame relative to $o_0$ frame, denoted as $T^0_2$, is given as:

$$T^0_2 = A_1A_2 = \begin{bmatrix} -Sq_1Cq_2 & -Cq_1 & Sq_1Sq_2 & -b_2Sq_1Cq_2 - a_1Sq_1 \\ Cq_1Cq_2 & -Sq_1 & -Cq_1Sq_2 & b_2Cq_1Cq_2 + a_1Cq_1 \\ Sq_2 & 0 & Cq_2 & b_2Sq_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (B.16)$$

Therefore $z_2$, $o_2$ are obtained from (B.16)

$$z_2 = \begin{bmatrix} Sq_1 \cdot Sq_2 \\ -Cq_1 \cdot Sq_2 \\ Cq_2 \end{bmatrix}, \quad o_2 = \begin{bmatrix} -b_2Sq_1Cq_2 - a_1Sq_1 \\ b_2Cq_1Cq_2 + a_1Cq_1 \\ b_2Sq_2 \end{bmatrix}. \quad (B.17)$$

**Manipulator Jacobian**

The Jacobian matrix $J_m \in \mathbb{R}^{6 \times 2}$ can be developed from $o_0$, $o_1$ and $o_2$ and $z_0$, $z_1$ and $z_2$. The Jacobian matrix $J_m$ is given as

$$J_m = \begin{bmatrix} z_0 \times (o_2 - o_0) \\ z_0 \end{bmatrix} \begin{bmatrix} z_1 \times (o_2 - o_1) \\ z_1 \end{bmatrix} \equiv \begin{bmatrix} J_{m_1} & J_{m_2} \\ J_{m_2} & J_{m_2} \end{bmatrix} = \begin{bmatrix} J_{v_2} \\ J_{\omega_2} \end{bmatrix} \quad (B.18)$$

where the first element of the first row in (B.18) can be calculated using the origins in (B.17) and (B.11) as

$$o_2 - o_0 = o_2 = \begin{bmatrix} -b_2Sq_1Cq_2 - a_1Sq_1 \\ b_2Cq_1Cq_2 + a_1Cq_1 \\ b_2Sq_2 \end{bmatrix}. \quad (B.19)$$
and the cross product is found to be

\[ J_{mv1} = z_0 \times (o_2 - o_0) = \begin{bmatrix} \hat{x} & \hat{y} & \hat{z} \\ 0 & 0 & 1 \\ -b_2 S_1 C_2 - a_1 S_1 & b_2 C_1 C_2 + a_1 C_1 & b_2 S_2 \end{bmatrix} \]

\[ = \begin{bmatrix} -b_2 C_1 C_2 - a_1 C_1 \\ -b_2 S_1 C_2 - a_1 S_1 \\ 0 \end{bmatrix} \] (B.20)

The second element of the first row vector of B.18 was calculated in a similar manner

\[ o_2 - o_1 = \begin{bmatrix} -b_2 S_1 C_2 \\ b_2 C_1 C_2 \\ b_2 S_2 \end{bmatrix}, \]

and

\[ J_{mv2} = z_1 \times (o_2 - o_1) = \begin{bmatrix} \hat{x} & \hat{y} & \hat{z} \\ C_1 & S_1 & 0 \\ -b_2 S_1 C_2 & b_2 C_1 C_2 & b_2 S_2 \end{bmatrix} \]

\[ = \begin{bmatrix} b_2 S_1 S_2 & -b_2 C_1 S_2 & b_2 C_2 \end{bmatrix}^T. \] (B.21)

Thus, \( J_m(q) \) in (B.18) is given by

\[ J_m = \begin{bmatrix} -b_2 C_1 C_2 - a_1 C_1 & b_2 S_1 S_2 \\ -b_2 S_1 C_2 - a_1 S_1 & -b_2 C_1 S_2 \\ 0 & b_2 C_2 \\ 0 & C_1 \\ 0 & S_1 \\ 1 & 0 \end{bmatrix}. \] (B.22)

**Translational Kinetic Energy From the Robot Manipulator**

The inertia term in the translational kinetic energy in B.4 can be represented as

\[ \sum_{i=1}^{2} (m_i J_{v_i}^T J_{v_i}). \] (B.23)

Rewriting the link origin,

\[ o_1 = \begin{bmatrix} -a_1 S_1 \\ a_1 C_1 \\ 0 \end{bmatrix}, \quad o_2 = \begin{bmatrix} -b_2 S_1 C_2 - a_1 S_1 \\ b_2 C_1 C_2 + a_1 C_1 \\ b_2 S_2 \end{bmatrix} \] (B.24)

Taking partial derivative of the origin \( o_1 \) yields

\[ \frac{d o_1}{dq} = \begin{bmatrix} \frac{\partial o_1}{\partial q_1} & \frac{\partial o_1}{\partial q_2} \\ \frac{\partial o_1}{\partial q_1} & \frac{\partial o_1}{\partial q_2} \end{bmatrix} = \begin{bmatrix} -a_1 C_1 & 0 \\ -a_1 S_1 & 0 \end{bmatrix} \equiv J_{v_1}. \] (B.25)
and the time derivative of the origin \( o_2 \) yields

\[
\frac{do_2}{dq} = \begin{bmatrix}
-b_2 Cq_1 Cq_2 - a_1 Cq_1 & b_2 Sq_1 Sq_2 \\
-b_2 Sq_1 Cq_2 - a_1 Sq_1 & -b_2 Cq_1 Sq_2 \\
0 & b_2 Cq_2 \\
\end{bmatrix} \equiv J_{v_2} \tag{B.26}
\]

where \( a_1 \) is expressed as the total length of the first link. Note that this time derivative exactly equals the Jacobian matrix \( J_{v_2} \) in (B.22) which proves the validity of the Jacobian matrix for the link. The inertia matrix given in (B.4), (B.25) and (B.26) are used to calculated the translational energy of the first link

\[
m_1 J_{v_1}^T J_{v_1} = m_1 \begin{bmatrix}
-l_1 Cq_1 & 0 \\
0 & l_1 Sq_1 \\
0 & 0
\end{bmatrix} \begin{bmatrix}
-l_1 Cq_1 & 0 \\
0 & -l_1 Sq_1 \\
0 & 0
\end{bmatrix} = \begin{bmatrix}
m_1 l_1^2 & 0 \\
0 & 0
\end{bmatrix}. \tag{B.27}
\]

Note that the term \( a_1 \) in the original Jacobian matrix was replaced with \( l_1 \) here to represent position of the center of mass in order to calculate the kinetic energy of the link. Likewise, the translational kinetic energy of the second link is calculated as:

\[
m_2 J_{v_2}^T J_{v_2} = m_2 \begin{bmatrix}
-l_2 Cq_1 Cq_2 - a_1 Cq_1 & -l_2 Sq_1 Cq_2 - a_1 Sq_1 & 0 \\
l_2 Sq_1 Cq_2 & -l_2 Cq_1 Sq_2 & l_2 Cq_2 \\
0 & l_2 Cq_2 & 0
\end{bmatrix} \cdot \begin{bmatrix}
l_2^2 C^2 q_2 + 2a_1 l_2 Cq_2 + a_1^2 & 0 \\
0 & l_2^2
\end{bmatrix} = \begin{bmatrix}
m_2 (l_2^2 C^2 q_2 + 2a_1 l_2 Cq_2 + a_1^2) & 0 \\
0 & m_2 l_2^2
\end{bmatrix}
\]

where similarly, \( b_2 \) was replaced by \( l_2 \) for kinetic energy calculation and the abbreviation \( C^2 q_2 = \cos^2(q_2) \) was used. Hence, the translational energy of the whole manipulator is

\[
I_{Kv_2} = J_{v_1}^T J_{v_1} + m_1 J_{v_2}^T J_{v_2} = \begin{bmatrix}
l_1^2 & 0 \\
0 & 0
\end{bmatrix} + \begin{bmatrix}
m_2 (l_2^2 C^2 q_2 + 2a_1 l_2 Cq_2 + a_1^2) & 0 \\
0 & m_2 l_2^2
\end{bmatrix} = \begin{bmatrix}
l_1^2 + m_2 (l_2^2 C^2 q_2 + 2a_1 l_2 Cq_2 + a_1^2) & 0 \\
0 & m_2 l_2^2
\end{bmatrix}.
\]

**Rotational Kinetic Energy**

In order to find the rotational kinetic energy which is given

\[
J^T_{\omega_1} P_{\omega_1}^0 J_{\omega_1}^0 J^T_{\omega_2}, \tag{B.28}
\]
the rotation matrix $R_0^1$ in (B.9) will be used and the Jacobian matrix of the first link, $J_{\omega_1}$, will be found from the angular velocity, $\omega_1$, through the skew-symmetric matrix [9]

$$S(\omega_1) \equiv R_0^1 R_0^{0T},$$

$$\omega_1 = J_{\omega_1} \cdot \dot{q}_1.$$

Differentiating $R_0^1$ yields

$$\dot{R}_0^1 = \left[\begin{array}{ccc}
-Cq_1 \dot{q}_1 & 0 & -Sq_1 \dot{q}_1 \\
-Sq_1 \dot{q}_1 & 0 & Cq_1 \dot{q}_1 \\
0 & 0 & 0
\end{array}\right]. \quad (B.29)$$

Thus,

$$S(\omega_1) = \dot{R}_0^1 \cdot R_0^{0T} = \left[\begin{array}{ccc}
-Cq_1 \dot{q}_1 & 0 & -Sq_1 \dot{q}_1 \\
-Sq_1 \dot{q}_1 & 0 & Cq_1 \dot{q}_1 \\
0 & 0 & 0
\end{array}\right] \left[\begin{array}{ccc}
-Sq_1 & Cq_1 & 0 \\
0 & 0 & 1 \\
Cq_1 & Sq_1 & 0
\end{array}\right]$$

$$= \left[\begin{array}{ccc}
Cq_1 Sq_1 \dot{q}_1 - Sq_1 Cq_1 \dot{q}_1 & Cq_1^2 \dot{q}_1 - Sq_1^2 \dot{q}_1 - \dot{q}_1 \\
\dot{q}_1 & Cq_1 Sq_1 \dot{q}_1 - Sq_1 Cq_1 \dot{q}_1 & 0 \\
0 & 0 & 0
\end{array}\right]$$

$$= \left[\begin{array}{ccc}
0 & -\dot{q}_1 & 0 \\
\dot{q}_1 & 0 & 0 \\
0 & 0 & 0
\end{array}\right].$$

Then, the above equation yields $\omega_1$ and $J_{\omega_1}$ as

$$\omega_1 = \left[\begin{array}{c}
0 \\
0 \\
\dot{q}_1
\end{array}\right] = \left[\begin{array}{ccc}
0 & 0 & 0 \\
0 & 0 & 1 \\
1 & 0
\end{array}\right] \left[\begin{array}{c}
\dot{q}_1 \\
\dot{q}_2
\end{array}\right]$$

$$\therefore \quad J_{\omega_1} = \left[\begin{array}{ccc}
0 & 0 \\
0 & 0 \\
1 & 0
\end{array}\right]. \quad (B.30)$$

The rotational kinetic energy due to the angular velocity $\omega_1$ can be expressed as

$$(J_{\omega_1}^T R_0^1 I_1 (R_0^1)^T J_{\omega_1}) = [(R_0^1)^T J_{\omega_1}]^T I_1 [(R_0^1)^T J_{\omega_1}] \quad (B.31)$$

where,

$$(R_0^1)^T J_{\omega_1} = \left[\begin{array}{ccc}
-Sq_1 & Cq_1 & 0 \\
0 & 0 & 1 \\
Cq_1 & Sq_1 & 0
\end{array}\right] \left[\begin{array}{ccc}
0 & 0 & 0 \\
0 & 1 & 0 \\
1 & 0
\end{array}\right] = \left[\begin{array}{ccc}
0 & 0 \\
0 & 1 & 0 \\
0 & 0
\end{array}\right]. \quad (B.32)$$
Thus, the energy equation becomes

\[ I_1 \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \end{bmatrix} = I_1 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \] (B.33)

where \( I_1 \) describes the moment of inertia in the center of mass of the first link. \( R_2^0 \) is obtained from \( T_2^0 \) as

\[ R_2^0 = \begin{bmatrix} -S_1 C_2 q_1 + C_1 q_1 & S_1 q_1 & S_1 q_2 q_1 + S_1 q_2 q_2 \\ C_1 C_2 q_1 - S_1 q_1 & -S_1 q_1 & -C_1 q_1 q_2 \\ S_2 q_2 & 0 & C_2 q_2 \end{bmatrix} \] (B.34)

Differentiating \( R_2^0 \) produces

\[ \dot{R}_2^0 = \begin{bmatrix} -C_1 C_2 q_1 + S_1 q_2 q_2 & S_1 q_1 & S_1 q_2 q_1 + S_1 q_2 q_2 \\ -S_1 C_2 q_1 + C_1 q_1 & -C_1 q_1 q_2 \\ S_1 q_2 q_2 & 0 & -S_2 q_2 q_2 \end{bmatrix} \] (B.35)

therefore

\[ \dot{R}_2^0 \cdot (R_2^0)^T = S(\omega_2) \] (B.36)

Thus,

\[ S(\omega_2)(1,1) = (C_1 C_2 q_1 - S_1 q_2 q_2)(S_1 C_2 q_1 - S_1 q_2 q_2) + (C_1 q_2 - S_1 q_2 q_2)(S_1 q_2 q_2) \]

\[ = (C_1 q_2 q_2 + C_1 q_2 q_2)(C_1 q_2 q_2 - S_1 q_2 q_2) + (C_1 q_2 q_2 - S_1 q_2 q_2)(S_1 q_2 q_2) \]

\[ = C_1 q_2 q_2 (C_1 q_2 q_2 - S_1 q_2 q_2) - S_1 q_2 q_2 C_1 q_2 q_2 \]

\[ = C_1 q_2 q_2 - S_1 q_2 q_2 C_1 q_2 q_2 \]

\[ = 0 \]

and

\[ S(\omega_2)(2,2) = S(\omega_2)(3,3) = 0. \]

Thus, \( \omega_2 \) is obtained as

\[ \omega_2(1) = (C_1^2 + S_1^2)C_1 q_2 = C_1 q_2 \]

\[ \omega_2(2) = (C_1 q_2 q_2 + S_2 q_2)(C_1 q_2 q_2 + S_2 q_2) + (C_1 q_2 q_2 + S_2 q_2)(C_2 q_2 q_2) \]

\[ = (C_1 q_2 q_2 + C_2 q_2 q_2)(C_1 q_2 q_2 + S_2 q_2) + S_1 (C_2 + S_2 q_2) \]
\[ \omega_2(3) = (S q_1 C q_2 \dot{q}_1 + C q_1 S q_2 \dot{q}_2) S q_1 C q_2 + C q_1^2 \dot{q}_1 + (S q_1 S q_2 \dot{q}_1 + C q_1 C q_2 \dot{q}_2) S q_1 S q_2 \]

\[ \omega_2 = \begin{bmatrix} C q_1 \dot{q}_2 \\ S q_1 \dot{q}_2 \\ \dot{q}_1 \end{bmatrix}. \] \hspace{1cm} (B.37)

From \( \omega_2 \),

\[ \omega_2 = \begin{bmatrix} C q_1 \dot{q}_2 \\ S q_1 \dot{q}_2 \\ \dot{q}_1 \end{bmatrix} = \begin{bmatrix} 0 & C q_1 \\ 0 & S q_1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} = J_{\omega_2} \dot{q}. \hspace{1cm} (B.38) \]

Note that the Jacobian matrix in the above equation exactly equals the Jacobian matrix \( J_{\omega_2} \) in (B.22). Kinetic Energy of the second link from the angular velocity \( \omega_2 \) is given by

\[ \left[(R_2^0)^T J_{\omega_2}\right]^T I_2 \left[(R_2^0)^T J_{\omega_2}\right] \]

and

\[ (R_2^0)^T J_{\omega_2} = \begin{bmatrix} -S q_1 C q_2 & C q_1 C q_2 & S q_2 \\ -C q_1 & -S q_1 & 0 \\ S q_1 S q_2 & -C q_1 S q_2 & C q_2 \end{bmatrix} \begin{bmatrix} 0 & C q_1 \\ 0 & S q_1 \\ 1 & 0 \end{bmatrix} \]

\[ = \begin{bmatrix} S q_2 & -S q_1 C q_2 C q_1 + S q_1 C q_2 C q_1 \\ 0 & -(C q_1^2 + S q_1^2) \\ C q_2 & S q_1 S q_2 C q_1 - S q_1 S q_2 C q_1 \end{bmatrix} \]

\[ = \begin{bmatrix} S q_2 & 0 \\ 0 & -1 \\ C q_2 & 0 \end{bmatrix}. \] \hspace{1cm} (B.39)

Then the kinetic energy of the second link yields

\[ \left[ \begin{bmatrix} S q_2 & 0 \\ 0 & -1 \\ 0 & 0 \end{bmatrix}\right] I_2 \left[ \begin{bmatrix} S q_2 & 0 \\ 0 & -1 \\ C q_2 & 0 \end{bmatrix}\right] = \left[ \begin{bmatrix} S^2 q_2 + C^2 q_2 & 0 \\ 0 & 1 \end{bmatrix}\right] = \left[ \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right]. \] \hspace{1cm} (B.40)

Thus,

\[ I_2 \left[ \begin{bmatrix} S q_2 & 0 & C q_2 \\ 0 & -1 & 0 \\ C q_2 & 0 \end{bmatrix}\right] = I_2 \left[ \begin{bmatrix} S^2 q_2 + C^2 q_2 & 0 \\ 0 & 1 \end{bmatrix}\right] = I_2 \left[ \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right]. \] \hspace{1cm} (B.41)
Hence, the kinetic energy term due to the angular velocity is

\[ I_{K\omega} = \sum_{i=1}^{2} J_{\omega_i}^T R_i^0 I_i (R_i^0)^T J_{\omega_i} \]

\[ = \left\{ J_{\omega_1}^T R_1^0 I_1 (R_1^0)^T J_{\omega_1} + J_{\omega_2}^T R_2^0 I_2 (R_2^0)^T J_{\omega_2} \right\} \]

\[ = I_1 \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + I_2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} I_1 + I_2 & 0 \\ 0 & I_2 \end{bmatrix}. \quad \text{(B.42)} \]

Hence the total inertia matrix in the kinetic energy equation is given by \( I_{Kv} \) and \( I_{K\omega} \) yielding

\[ M_m(q) = \sum_{i=1}^{2} \left\{ m_i J_{v_i}^T J_{v_i} + [R_i^0 J_{\omega_i}]^T I_i [R_i^0 J_{\omega_i}] \right\} \]

\[ = \begin{bmatrix} ml_1^2 + m_2(l_2^2 C_{q_2}^2 + 2a_1 l_2 C q_2 + a_1^2) & 0 \\ 0 & m_2 l_2^2 \end{bmatrix} + \begin{bmatrix} I_1 + I_2 & 0 \\ 0 & I_2 \end{bmatrix} \]

\[ = \begin{bmatrix} ml_1^2 + m_2(l_2^2 C_{q_2}^2 + 2a_1 l_2 C q_2 + a_1^2) + I_1 + I_2 & 0 \\ 0 & m_2 l_2^2 + I_2 \end{bmatrix} \]

\[ = \begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix} \quad \text{(B.43)} \]

where

\[ d_{11} = ml_1^2 + m_2(l_2^2 C_{q_2}^2 + 2a_1 l_2 C q_2 + a_1^2) + I_1 + I_2 \]

\[ d_{12} = d_{21} = 0 \]

\[ d_{22} = m_2 l_2^2 + I_2. \quad \text{(B.44)} \]

Rewriting the dynamics,

\[ D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau. \]

From the Euler-Lagrangian equations of motion for n-link robot manipulator ??

\[ \sum_{n}^{j=1} d_{kj}(q)\ddot{q}_j + \sum_{n}^{i=1} \sum_{n}^{j=1} C_{ijk}(q)q_i \dot{q}_i \dot{q}_j + G_k(q) = \tau_k; k = 1, 2, \ldots, n \quad \text{(B.45)} \]

where \( n = 2 \), the first term in the above equation is calculated as

\[ \sum_{n}^{j=1} d_{kj}(q)\ddot{q}_j. \]

\[ k = 1, \sum_{2}^{j=1} d_{ij}(q)\ddot{q}_j = d_{11}(q)\ddot{q}_1 + d_{12}(q)\ddot{q}_2 \]

\[ = [ml_1^2 + m_2(l_2^2 C_{q_2}^2 + 2a_1 l_2 C q_2 + a_1^2) + I_1 + I_2] \ddot{q}_1 \]

\[ k = 2, \sum_{2}^{j=1} d_{2j}(q)\ddot{q}_j = d_{21}(q)\ddot{q}_1 + d_{22}(q)\ddot{q}_2 = (m_2 l_2^2 + I_2) \ddot{q}_2 \]
where (B.44) was used. The Coriolis-Centripedal term can be calculated from the general form \( \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} C_{ijk}(q) \dot{q}_i \dot{q}_j \) when \( k=1,2 \) as follows

\[
\begin{align*}
  k &= 1, \quad \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} C_{ij1}(q) \dot{q}_i \dot{q}_j = \sum_{i=1}^{\infty} (C_{i11}(q) \dot{q}_i \dot{q}_j + C_{i21}(q) \dot{q}_i \dot{q}_j) \\
  k &= 2, \quad \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} C_{ij2}(q) \dot{q}_i \dot{q}_j = \sum_{i=1}^{\infty} (C_{i12}(q) \dot{q}_i \dot{q}_j + C_{i22}(q) \dot{q}_i \dot{q}_j)
\end{align*}
\]

where for \( k = 1 \),

\[
\begin{align*}
  C_{111} &= \frac{1}{2} \frac{\partial d_{11}}{\partial q_1} = 0 \\
  C_{121} &= C_{211} = \frac{1}{2} \frac{\partial d_{11}}{\partial q_2} = -m_2 l_2^2 C q_2 \dot{q}_2 - m_2 a_1 l_2 S q_2 \dot{q}_2 \\
  C_{221} &= \frac{1}{2} \frac{\partial d_{12}}{\partial q_2} - \frac{1}{2} \frac{\partial d_{22}}{\partial q_1} = 0.
\end{align*}
\]

and for \( k = 2 \),

\[
\begin{align*}
  C_{112} &= \frac{1}{2} \frac{\partial d_{21}}{\partial q_1} - \frac{1}{2} \frac{\partial d_{11}}{\partial q_2} = m_2 l_2^2 C q_2 \dot{q}_2 + m_2 a_1 l_2 S q_2 \dot{q}_2 \\
  C_{122} &= C_{212} = \frac{1}{2} \frac{\partial d_{22}}{\partial q_1} = 0 \\
  C_{222} &= \frac{1}{2} \frac{\partial d_{22}}{\partial q_2} = 0.
\end{align*}
\]

The potential energy, which is needed in the Lagrangian, can be defined as

\[
G_k = \frac{\partial P_m}{\partial q}, \quad P_m = \sum_{i=1}^{2} P_i = P_1 + P_2 \tag{B.46}
\]

where \( P_m \) is the sum of the potential energy \( P_1 \) and \( P_2 \) of each link. The potential energy is calculated as

\[
\begin{align*}
  P_1 &= m_1 g l_1 C q_1, \\
  P_2 &= m_2 g (l_2 C q_1 C q_2 + a_1 C q_1), \\
  P_m &= m_1 g l_1 C q_1 + m_2 g (l_2 C q_1 C q_2 + a_1 C q_1).
\end{align*}
\]

Then
\[ G_1 = \frac{\partial P_m}{\partial q_1} = -m_1 gl_1 S q_1, \quad G_2 = \frac{\partial P_m}{\partial q_2} = -m_2 gl_2 C q_1 S q_2. \]

Substituting the above equations into the equations of motion for the two-link robot manipulator yields

\[
d_{11}(q) \ddot{q}_1 + c_{121}(q) \dot{q}_2 \dot{q}_1 + c_{211}(q) \dot{q}_1 \dot{q}_2 + G_1(q) = \tau_1 \\
d_{22}(q) \ddot{q}_2 + c_{112}(q) \dot{q}_1 \dot{q}_1 + G_2(q) = \tau_2
\]

Converting the above equation into a single matrix form yields

\[
\begin{bmatrix} d_{11}(q) & 0 \\ 0 & d_{22}(q) \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} c_{121}(q) \dot{q}_2 & c_{211}(q) \dot{q}_1 \\ c_{112}(q) \dot{q}_1 & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} + \begin{bmatrix} G_1(q) \\ G_2(q) \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}.
\]

Thus,

\[
M_m(q) \ddot{q} + C_m(q, \dot{q}) \dot{q} + G_m(q) = \tau_m
\]

where

\[
M_m(q) = \begin{bmatrix} d_{11}(q) & 0 \\ 0 & d_{22}(q) \end{bmatrix}, \quad C_m(q, \dot{q}) \dot{q} = \begin{bmatrix} c_{121}(q) \dot{q}_2 & c_{211}(q) \dot{q}_1 \\ c_{112}(q) \dot{q}_1 & 0 \end{bmatrix}, \\
G_m(q) = \begin{bmatrix} G_1(q) \\ G_2(q) \end{bmatrix}, \quad \tau_m = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}.
\]
Appendix C
Rotations of the UAVRM

1. Euler Angles (Roll, Pitch, and Yaw) Rotation Matrix

The rotation matrix, \( R_F(\Theta) \), transforms quantity from the body-fixed to the earth-fixed frame, rotating roll, pitch, and yaw motion in order as shown in (2.2) or (4.5). This can be decomposed into three rotations as follows.

Rotation Matrix about \( x \)-axis

The rotation by angle \( \phi \) about \( x \)-axis in 3D space as shown in Figure C.1 yields the rotation matrix \( R_{x,\phi} \)

\[
R_{x,\phi} = \begin{bmatrix}
    x_1 \cdot x_0 & y_1 \cdot x_0 & z_1 \cdot x_0 \\
    x_1 \cdot y_0 & y_1 \cdot y_0 & z_1 \cdot y_0 \\
    x_1 \cdot z_0 & y_1 \cdot z_0 & z_1 \cdot z_0
\end{bmatrix} = \begin{bmatrix}
    1 & 0 & 0 \\
    0 & \cos \phi & -\sin \phi \\
    0 & \sin \phi & \cos \phi
\end{bmatrix} \in SO(3). \quad (C.1)
\]

From the Figure C.1 we can see that the rotation is about \( x \)-axis, therefore
Figure C.2 Rotation ($\theta$) Matrix about $Y$-axis
\[ x_1 \cdot x_0 = |x_1| |x_0| \cos 0^\circ = 1, \]

Note that \(|\cdot| = 1\) is used here since all the vectors are unit vectors.

\[ y_1 \text{ and } z_1 \text{ are rotated from } y_0 \text{ and } z_0 \text{ by angle } \phi \text{ respectively, therefore} \]

\[ y_1 \cdot y_0 = |y_1| |y_0| \cos \phi = \cos \phi \]
\[ z_1 \cdot z_0 = |z_1| |z_0| \cos \phi = \cos \phi \]

Checking the angle between other axes, it can also be obtained that

\[ y_1 \cdot z_0 = |y_1| |z_0| \cos \left( \frac{\pi}{2} - \phi \right) = \sin \phi, \]
\[ z_1 \cdot y_0 = |z_1| |y_0| \cos \left( \frac{\pi}{2} + \phi \right) = -\sin \phi \]

Note that \(x_1\) is perpendicular to \(y_1\) and \(z_1\), and \(x_0\) is also perpendicular to \(y_1\) and \(z_1\), which yields

\[ x_1 \cdot y_1 = x_1 \cdot z_1 = 0 \]
\[ y_1 \cdot x_0 = z_1 \cdot x_0 = 0 \]

Thus, the result in (C.1) is verified.

**Rotation Matrix about y-axis**

In a similar fashion of developing the rotation matrix about \(x\)-axis, the rotation matrix about \(y\)-axis by an angle of \(\theta\) is developed as

\[
R_{y,\theta} = \begin{bmatrix}
 x_1 \cdot x_0 & 0 & z_1 \cdot x_0 \\
 0 & 1 & 0 \\
 x_1 \cdot z_0 & 0 & z_1 \cdot z_0
\end{bmatrix} = \begin{bmatrix}
 \cos \theta & 0 & \sin \theta \\
 0 & 1 & 0 \\
 -\sin \theta & 0 & \cos \theta
\end{bmatrix} \in SO(3) \quad (C.2)
\]

where the perpendicular parts are already made zero. Figure C.2 shows the rotation about \(y\) by an angle \(\theta\), which yields

\[ x_1 \cdot x_0 = |x_1| |x_0| \cos \theta = \cos \theta \]
\[ x_1 \cdot z_0 = |x_1| |z_0| \cos \left( \frac{\pi}{2} + \theta \right) = -\sin \theta \]
Figure C.3 Rotation $(\psi)$ Matrix about $Z$-axis
and
\[ z_1 \cdot x_0 = |z_1||x_0| \cos \left( \frac{\pi}{2} - \theta \right) = \sin \theta \]
\[ z_1 \cdot z_0 = |z_1||z_0| \cos \theta = \cos \theta. \]

**Rotation Matrix about z-axis**

Similarly, the rotation about z-axis by an angle \( \psi \) as shown in Figure C.3 results in
\[
R_{z,\psi} = \begin{bmatrix}
  x_1 \cdot x_0 & y_1 \cdot x_0 & 0 \\
  x_1 \cdot y_0 & y_1 \cdot y_0 & 0 \\
  0 & 0 & z_1 \cdot z_0
\end{bmatrix} = \begin{bmatrix}
  \cos \psi & -\sin \psi & 0 \\
  \sin \psi & \cos \psi & 0 \\
  0 & 0 & 1
\end{bmatrix} \in SO(3) \tag{C.3}
\]

where \( z_1 \cdot x_0, z_1 \cdot y_0, x_1 \cdot z_0 \), and \( y_1 \cdot z_0 \) are perpendicular to each other, and
\[
x_1 \cdot x_0 = \cos \psi
\]
\[
x_1 \cdot y_0 = \cos \left( \frac{\pi}{2} - \psi \right) = \sin \psi,
\]
and
\[
y_1 \cdot x_0 = \cos \left( \frac{\pi}{2} + \psi \right) = -\sin \psi
\]
\[
y_1 \cdot y_0 = \cos \psi.
\]

2. **UAVRM Total System Coordinate Frames and Rotations**

In this section, the forward kinematics of the UAVRM system will be developed using D-H convention and homogeneous transformation. In the development of the kinematics of the system, the joint configuration is described to show the rotations and translations made by each joints.

1. **UAVRM System Configuration**

The relationship shown in Figure C.4 describes the assignment of Denavit-Hartenberg coordinate frame for the integrated UAVRM system. The corresponding Denavit-Hartenberg Table for the UAVRM system is shown in Figure C.5. The homogeneous equations in (4.6), (4.7) and (4.8) corresponding to the first, second and third link of the UAVRM system are shown in Figure C.6, C.7 and C.8 respectively.
Figure C.4 UAVRM Integrated System Configuration

<table>
<thead>
<tr>
<th>Link(i)</th>
<th>$\theta_{x,y,z}^{z_{i-1}}$</th>
<th>d</th>
<th>a / b</th>
<th>$\alpha_{x,y_{z_{i-1}}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$(\phi/\theta) + 90^\circ$</td>
<td>$d_1$</td>
<td>0</td>
<td>$+90^\circ$</td>
</tr>
<tr>
<td>2</td>
<td>$q_1 + 90^\circ$</td>
<td>0</td>
<td>$a_2$</td>
<td>$+90^\circ$</td>
</tr>
<tr>
<td>3</td>
<td>$q_2 + 0^\circ$</td>
<td>$\phi$</td>
<td>$b_3$</td>
<td>$-90^\circ$</td>
</tr>
</tbody>
</table>

Figure C.5
Figure C.6 Homogeneous Transformation Matrix: $A_1$ for the First Link

\[ A_1 = R_{ZYX} \cdot R_{z, \theta} \cdot T_{z, d_1} \cdot R_{x, \alpha} \]
Figure C.7 Homogeneous Transformation Matrix: $A_2$ for the Second Link

- $R_{z, \theta = 90^\circ}$: Rotation $\theta = +90^\circ$ about z-axis
- $T_{x, \alpha_2}$: Translation $\alpha_2$ along x-axis
- $R_{x, \alpha} = +90^\circ$: Rotation $\alpha = +90^\circ$ about x-axis
- $A_2 = R_{z, \theta = 90^\circ} \cdot T_{x, \alpha_2} \cdot R_{x, \alpha = 180^\circ}$
Figure C.8 Homogeneous Transformation Matrix: \( A_3 \) for the Third Link


