

LQR Control for a Mono-Propelled VTOL Drone

Logan Dapp^{*}, Nicholas E. Hirsch[†], Dylan L. Duenas[‡], Steven Kinsell[§], Isabella D. Fernandez[¶], and Ethan Reinhart^{||}
University of Florida, Gainesville, FL, 32611, United States

As autonomous landing rockets become more common in industry, Florida Rocket Lab needs a testbed to develop the guidance, navigation, and controls for their propulsive landing vehicle *Hopper*. A mono-propelled vertical take-off and landing drone *Skipper* is an optimal choice as it is relatively safe, cost-effective, and mirrors the dynamics of a rocket closely. This paper first covers the derivation of the equations of motion for a mono-propelled VTOL drone. Using these equations, a linear state space can be constructed and controlled using a linear-quadratic regulator. Due to the highly nonlinear nature of the system, a gain-scheduling program is then implemented based on the error between the linear and nonlinear states. The combined control system is then implemented in MATLAB/Simulink and used to control the drone in a variety of environments. This paper aims to guide other student teams in the development of similar projects.

I. Introduction

Autonomous landing rockets are becoming more common in current industry, such as Blue Origin's *New Shepard* and SpaceX's *Falcon 9* and *Falcon Heavy* rockets. With this increase in autonomous vertical landing vehicles in industry, it would benefit students to develop an understanding of the dynamics and guidance, navigation, and controls (GNC) of a vertical take-off and landing (VTOL) vehicle. *Hopper* is a student-developed liquid rocketry project founded by *Florida Rocket Lab* at the University of Florida. To develop and test the GNC of this propulsive landing vehicle, the choice of a mono-propelled VTOL drone is an appropriate pilot project. The dynamics of the drone, *Skipper*, closely mirrors that of a rocket while also being both economically viable and safe for a student organization. The *Skipper* project aims to give undergraduate students experience with GNC, in addition to serving as a testbed for *Hopper*. This paper outlines the dynamics of the system, the construction and control of a linear state space with the use of a linear-quadratic regulator (LQR), and the implementation of these into simulations with Simulink. These findings can contribute to improving the controls of autonomous landing systems, especially those with dynamics similar to that of *Skipper*. Moreover, the findings can offer a framework and entry point for other student organizations with similar projects.

II. System Dynamics

Skipper is a mono-propelled VTOL drone with its propulsive capability, two contra-rotating propellers, located at the bottom of the vehicle. It is modeled as a rigid cylinder of diameter, D , height, H , and mass M in three-dimensional Euclidean space, \mathbb{E}^3 with 6 degrees-of-freedom (DOF) [1, 2] — three translational and three rotational. To ensure full controllability, there are three methods of control the drone may use:

- 1) Thrust generated by throttling the propellers.
- 2) A 2-DOF gimbal, allowing for thrust vectoring.
- 3) Torque about the longitudinal axis generated from the relative angular rates of the two contra-rotating propellers.

Taken together, these controls allow control effort to be exerted along all six of *Skipper*'s DOF.

^{*}Mechanical Engineering and Computer Science Undergraduate and Florida Rocket Lab Guidance Navigation and Controls Team Member, Departments of Mechanical and Aerospace Engineering and Computer and Information Science and Engineering, AIAA Student Member 1822599.

[†]Aerospace Engineering Undergraduate and Florida Rocket Lab Guidance Navigation and Controls Team Director, Department of Mechanical and Aerospace Engineering, AIAA Student Member, 1810373.

[‡]Aerospace Engineering Undergraduate and Florida Rocket Lab Guidance Navigation and Controls Team Member, Department of Mechanical and Aerospace Engineering, AIAA Student Member 1810988.

[§]Mechanical Engineering Undergraduate and Florida Rocket Lab Guidance Navigation and Controls Team Member, Department of Mechanical and Aerospace Engineering, AIAA Student Member, 1823410.

[¶]Aerospace Engineering Undergraduate and Florida Rocket Lab Guidance Navigation and Controls Team Member, Department of Mechanical and Aerospace Engineering, AIAA Student Member, 1799497.

^{||}Aerospace Engineering Undergraduate and Florida Rocket Lab Guidance Navigation and Controls Team Member, Department of Mechanical and Aerospace Engineering, AIAA Student Member, 1812027.

A. Reference Frames

Consider an inertial frame, \mathcal{I} , fixed to the Earth with an associated right-handed basis $\{\hat{\mathbf{x}}_e, \hat{\mathbf{y}}_e, \hat{\mathbf{z}}_e\} \in \mathcal{I}$. In contrast to convention, we take $\hat{\mathbf{x}}_e$ to point straight up, normal to the $\hat{\mathbf{y}}_e\hat{\mathbf{z}}_e$ -plane corresponding to the ground. $\hat{\mathbf{z}}_e$ is not chosen to point straight down because doing so would induce gimbal lock in the starting configuration. Next, let \mathcal{U} be another reference frame fixed to *Skipper* with corresponding basis $\{\hat{\mathbf{x}}_b, \hat{\mathbf{y}}_b, \hat{\mathbf{z}}_b\} \in \mathcal{U}$. Assume that the two frames are aligned by default, and that as \mathcal{U} rotates with respect to \mathcal{I} this rotation may be described using a 3-2-1 intrinsic Tait-Bryan angle sequence, $\{\psi, \theta, \phi\}$, referred to as yaw, pitch, and roll respectively. This series of rotations is given by the following three direction-cosine matrices (DCM),

$$\mathbf{C}_3(\psi) = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{C}_2(\theta) = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix}, \quad \mathbf{C}_1(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix}. \quad (1)$$

The combined transformation from \mathcal{I} to \mathcal{U} is then given by,

$$\begin{aligned} \mathbf{C}_{\mathcal{I}}^{\mathcal{U}}(\psi, \theta, \phi) &= \mathbf{C}_1(\phi)\mathbf{C}_2(\theta)\mathbf{C}_3(\psi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (2) \\ &= \begin{bmatrix} \cos \psi \cos \theta & \cos \theta \sin \psi & -\sin \theta \\ \cos \psi \sin \phi \sin \theta - \cos \phi \sin \psi & \cos \phi \cos \psi + \sin \phi \sin \psi \sin \theta & \cos \theta \sin \phi \\ \sin \phi \sin \psi + \cos \phi \cos \psi \sin \theta & \cos \phi \sin \psi \sin \theta - \cos \psi \sin \phi & \cos \phi \cos \theta \end{bmatrix}. \quad (3) \end{aligned}$$

To make simply later derivations, we also introduce a frame fixed to the gimbal, \mathcal{G} , with a corresponding basis $\{\hat{\mathbf{x}}_g, \hat{\mathbf{y}}_g, \hat{\mathbf{z}}_g\} \in \mathcal{G}$. As previously stated, the gimbal has 2-DOF and in this case \mathcal{G} rotates with respect to \mathcal{U} through a 2-3 sequence represented by the angles, $\{\xi, \zeta\}$. Using Eq. (1) as a basis, we may form the \mathcal{U} to \mathcal{G} DCM,

$$\mathbf{C}_{\mathcal{U}}^{\mathcal{G}}(\xi, \zeta) = \mathbf{C}_3(\zeta)\mathbf{C}_2(\xi) = \begin{bmatrix} \cos \zeta & \sin \zeta & 0 \\ -\sin \zeta & \cos \zeta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \xi & 0 & -\sin \xi \\ 0 & 1 & 0 \\ \sin \xi & 0 & \cos \xi \end{bmatrix}. \quad (4)$$

To utilize a DCM, consider an arbitrary vector, \mathbf{v} , defined in an arbitrary frame, \mathcal{A} . We may denote this as ${}^{\mathcal{A}}\mathbf{v}$. Now consider another frame, \mathcal{B} , along with a DCM, $\mathbf{C}_{\mathcal{A}}^{\mathcal{B}}$, which converts between the two frames. To redefine ${}^{\mathcal{A}}\mathbf{v}$ in \mathcal{B} we use the the relation,

$${}^{\mathcal{B}}\mathbf{v} = \mathbf{C}_{\mathcal{A}}^{\mathcal{B}} {}^{\mathcal{A}}\mathbf{v}. \quad (5)$$

Additionally all DCMs are orthogonal, meaning they satisfy the following property:

$$\mathbf{C}\mathbf{C}^{-1} = \mathbf{C}\mathbf{C}^{\top} = \mathbf{I}, \quad (6)$$

where \mathbf{I} is the identity matrix; therefore, to convert ${}^{\mathcal{B}}\mathbf{v}$ back to \mathcal{A} we may use Eq. (6) to write

$${}^{\mathcal{A}}\mathbf{v} = (\mathbf{C}_{\mathcal{A}}^{\mathcal{B}})^{\top} {}^{\mathcal{B}}\mathbf{v} = \mathbf{C}_{\mathcal{B}}^{\mathcal{A}} {}^{\mathcal{B}}\mathbf{v}. \quad (7)$$

B. Kinematics

Let \mathbf{r} represent the displacement of *Skipper*'s center of gravity in \mathbb{E}^3 from which it lifts off, O , fixed in \mathcal{I} . We may write \mathbf{r} as

$${}^{\mathcal{I}}\mathbf{r} = x\hat{\mathbf{x}}_e + y\hat{\mathbf{y}}_e + z\hat{\mathbf{z}}_e, \quad (8)$$

where (x, y, z) represent the distance traveled from O along their respective basis vectors. Differentiating with respect to time, t , in \mathcal{I} , we obtain

$${}^{\mathcal{I}}\mathbf{v} = \dot{x}\hat{\mathbf{x}}_e + \dot{y}\hat{\mathbf{y}}_e + \dot{z}\hat{\mathbf{z}}_e. \quad (9)$$

However, it often useful to instead observe *Skipper*'s motion from \mathcal{U} , and as such we may write Eq (9) as

$${}^{\mathcal{U}}\mathbf{v} = u\hat{\mathbf{x}}_b + v\hat{\mathbf{y}}_b + w\hat{\mathbf{z}}_b, \quad (10)$$

where (u, v, w) represent the components of \mathbf{v} along their accompanying basis vector as discussed in [3]. To differentiate Eq (10), we make use of the transport theorem—which facilitates a time derivative in a non-inertial frame in terms of an inertial frame—so that we may use the Newton-Euler equations of motion, which are only applicable in inertial frames.

$$\frac{{}^{\mathcal{B}}d}{{}^{\mathcal{A}}dt} \mathcal{A}\mathbf{v} = \frac{{}^{\mathcal{A}}d}{{}^{\mathcal{A}}dt} \mathcal{A}\mathbf{v} + \boldsymbol{\omega} \times \mathcal{A}\mathbf{v} = \mathcal{A}\dot{\mathbf{v}} + \boldsymbol{\omega} \times \mathcal{A}\mathbf{v}. \quad (11)$$

Using Eq. (11) on Eq. (10) we may write

$${}^{\mathcal{I}}\mathbf{a} = {}^{\mathcal{U}}\dot{\mathbf{v}} + {}^{\mathcal{U}}\boldsymbol{\omega} \times \mathcal{U}\mathbf{v} \quad (12)$$

$$= \dot{u}\hat{\mathbf{x}}_b + \dot{v}\hat{\mathbf{y}}_b + \dot{w}\hat{\mathbf{z}}_b + (p\hat{\mathbf{x}}_b + q\hat{\mathbf{y}}_b + r\hat{\mathbf{z}}_b) \times (u\hat{\mathbf{x}}_b + v\hat{\mathbf{y}}_b + w\hat{\mathbf{z}}_b) \quad (13)$$

$$= \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} + \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} + \begin{bmatrix} qw - rv \\ ru - pw \\ pv - qu \end{bmatrix}, \quad (14)$$

where ${}^{\mathcal{U}}\boldsymbol{\omega}$ is the angular velocity of \mathcal{U} with respect to \mathcal{I} and (p, q, r) are its components written in $\{\hat{\mathbf{x}}_b, \hat{\mathbf{y}}_b, \hat{\mathbf{z}}_b\}$ [1, 3]. Note that in Eq. (14) we make use of the fact that

$$({}^{\mathcal{U}}\boldsymbol{\omega} \times) = {}^{\mathcal{U}}\boldsymbol{\omega}^\times = \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix}, \quad (15)$$

where $(\cdot)^\times$ represents the skew operator [1].

Because *Skipper* is a 6-DOF system, we need angular kinematics as well as the linear kinematics to fully define *Skipper*'s equations of motion (EOM). For angular kinematics, we define the angular momentum \mathbf{h} about *Skipper*'s center of mass (CM) to be

$${}^{\mathcal{U}}\mathbf{h} = \mathbf{J} {}^{\mathcal{U}}\boldsymbol{\omega}, \quad (16)$$

where \mathbf{J} is *Skipper*'s corresponding moment of inertia tensor. By defining \mathbf{J} in \mathcal{U} , the principal axes of a cylinder correspond to $\{\hat{\mathbf{x}}_b, \hat{\mathbf{y}}_b, \hat{\mathbf{z}}_b\}$, meaning

$$J_{xy} = J_{yx} = J_{xz} = J_{zx} = J_{yz} = J_{zy} = 0. \quad (17)$$

and hence \mathbf{J} reduces to

$$\mathbf{J} = J_{xx}\hat{\mathbf{x}}_b + J_{yy}\hat{\mathbf{y}}_b + J_{zz}\hat{\mathbf{z}}_b. \quad (18)$$

In matrix form, Eq. (16) may be rewritten as,

$${}^{\mathcal{U}}\mathbf{h} = \begin{bmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} J_{xx}p \\ J_{yy}q \\ J_{zz}r \end{bmatrix}. \quad (19)$$

Differentiating Eq. (19) in \mathcal{I} , and again applying Eq. (11), we find that,

$${}^{\mathcal{I}}\frac{d}{{}^{\mathcal{A}}dt} {}^{\mathcal{U}}\mathbf{h} = {}^{\mathcal{U}}\dot{\mathbf{h}} + {}^{\mathcal{U}}\boldsymbol{\omega} \times \mathcal{U}\mathbf{h} \quad (20)$$

$$= \begin{bmatrix} J_{xx}\dot{p} \\ J_{yy}\dot{q} \\ J_{zz}\dot{r} \end{bmatrix} + \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \begin{bmatrix} J_{xx}p \\ J_{yy}q \\ J_{zz}r \end{bmatrix} = \begin{bmatrix} J_{xx}\dot{p} + qr(J_{zz} - J_{yy}) \\ J_{yy}\dot{q} + pr(J_{xx} - J_{zz}) \\ J_{zz}\dot{r} + pq(J_{yy} - J_{xx}) \end{bmatrix}. \quad (21)$$

C. Kinetics

With the kinematics found via Eq. (14) and Eq. (21) we can now compute the EOM. Because *Skipper* is a 6-DOF rigid body, we must use Newton-Euler's laws for rigid bodies. To begin, consider the following actions, consisting of forces and pure torques, acting on *Skipper*:

- 1) Weight, ${}^I \mathbf{W} = -mg\hat{\mathbf{x}}_e$, where m is *Skipper*'s mass and g the acceleration due to gravity.
- 2) Thrust, ${}^G \mathbf{T} = T\hat{\mathbf{x}}_g$, where $T = \|\mathbf{T}\|$.
- 3) Reaction torque, approximated about the body, ${}^U \boldsymbol{\tau}_r = \tau_r\hat{\mathbf{x}}_u$, where $\tau_r = \|\boldsymbol{\tau}_r\|$.

Newton-Euler's first equation is,

$$\sum_i \mathbf{F}_i = m\mathbf{a}, \quad (22)$$

where $\{\mathbf{F}_i\}$ are the forces acting on the body and \mathbf{a} is the next acceleration. Using Eq. (14) in conjunction with the forces acting on the system, Eq. (22) expands to:

$$\mathbf{T} + \mathbf{W} = m {}^U \mathbf{a}. \quad (23)$$

However, all three vector quantities are in different frames and as such we make use of Eq. (3) and Eq. (4) in conjunction with Eq. (6) to convert all quantities to \mathcal{U} to obtain,

$$\mathbf{C}_{\mathcal{G}}^{\mathcal{U}} \begin{bmatrix} T \\ 0 \\ 0 \end{bmatrix} + \mathbf{C}_{\mathcal{I}}^{\mathcal{U}} \begin{bmatrix} -mg \\ 0 \\ 0 \end{bmatrix} = m \left(\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} + \begin{bmatrix} qw - rv \\ ru - pw \\ pv - qu \end{bmatrix} \right). \quad (24)$$

Moving onto Newton-Euler's second law, it is given by,

$$\sum_i (\mathbf{r}_i \times \mathbf{F}_i) + \sum_i \boldsymbol{\tau}_i = \frac{d}{dt} \mathbf{h}, \quad (25)$$

where \mathbf{r}_i is the lever arm (from the CM of the body) of the i -th force and $\{\boldsymbol{\tau}_i\}$ is the pure torques acting on the body. Using Eq. (21) in conjunction with the forces and pure torques acting on the system, Eq. (25) becomes,

$$\boldsymbol{\rho} \times \mathbf{T} + \boldsymbol{\tau}_r = \frac{d}{dt} {}^U \mathbf{h}. \quad (26)$$

$\boldsymbol{\rho} = -\rho\hat{\mathbf{x}}_b$ is the lever arm from the CM to the center of thrust. Again converting all the frames to, \mathcal{U} , the above can be rewritten as,

$$\begin{bmatrix} -\rho \\ 0 \\ 0 \end{bmatrix} \times \mathbf{C}_{\mathcal{G}}^{\mathcal{U}} \begin{bmatrix} T \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} \tau_r \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} J_{xx}\dot{p} + qr(J_{zz} - J_{yy}) \\ J_{yy}\dot{q} + pr(J_{xx} - J_{zz}) \\ J_{zz}\dot{r} + pq(J_{yy} - J_{xx}) \end{bmatrix}. \quad (27)$$

Combined, Eq. (24) and Eq. (27) form a system of 6 first-order ordinary differential equations (ODEs) which may be solved for $(\dot{u}, \dot{v}, \dot{w}, \dot{p}, \dot{q}, \dot{r})$ to obtain,

$$\dot{u} = (T \cos \xi \cos \zeta - mqw + mrv - mg \cos \psi \cos \theta / m), \quad (28)$$

$$\dot{v} = (T \sin \zeta + mpw - mru + mg \cos \phi \sin \psi - mg \cos \psi \sin \phi \sin \theta) / M, \quad (29)$$

$$\dot{w} = -(T \cos \zeta \sin \xi + mpv - mqu + mg \sin \phi \sin \psi + mg \cos \phi \cos \psi \sin \theta) / m, \quad (30)$$

$$\dot{p} = (\tau_r + J_{yy}qr - J_{zz}qr) / J_{xx}, \quad (31)$$

$$\dot{q} = -(J_{xx}pr - J_{zz}pr + T\rho \cos \zeta \sin \xi) / J_{yy}, \quad (32)$$

$$\dot{r} = -(T\rho \sin \zeta - J_{xx}pq + J_{yy}pq) / J_{zz}. \quad (33)$$

Eqs. (28 - 33) form a system of 6 equations and 6 unknowns, where (T, τ_r, ξ, ζ) are inputs to the system assumed to be known.

D. Position and Attitude Representation

To fully model the flight dynamics of *Skipper* it is necessary to introduce representations for the components of the position, (x, y, z) , in $\{\hat{\mathbf{x}}_e, \hat{\mathbf{y}}_e, \hat{\mathbf{z}}_e\}$ along with attitude, (ϕ, θ, ψ) . The former can simply be found by introducing a relation between ${}^I \mathbf{v}$ and ${}^U \mathbf{v}$ via Eq. (3) as,

$${}^I \mathbf{v} = (\mathbf{C}_{\mathcal{U}}^I)^T {}^U \mathbf{v}, \quad (34)$$

or alternatively,

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \mathbf{C}_{\mathcal{U}}^T \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix}, \quad (35)$$

which may be solved for $(\dot{x}, \dot{y}, \dot{z})$ to be,

$$\dot{x} = w \sin \phi \sin \psi + \cos \phi \cos \psi \sin \theta - v \cos \phi \sin \psi - \cos \psi \sin \phi \sin \theta + u \cos \psi \cos \theta, \quad (36)$$

$$\dot{y} = v \cos \phi \cos \psi + \sin \phi \sin \psi \sin \theta - w \cos \psi \sin \phi - \cos \phi \sin \psi \sin \theta + u \cos \theta \sin \psi, \quad (37)$$

$$\dot{z} = w \cos \phi \cos \theta - u \sin \theta + v \cos \theta \sin \phi. \quad (38)$$

A similar relation may be developed for relating $(\dot{\phi}, \dot{\theta}, \dot{\psi})$ to (p, q, r) , although because of the intrinsic nature of the 3-2-1 Tait-Bryant rotation sequence each of (ϕ, θ, ψ) are in different intermediate frames. Thus, we may solve for (p, q, r) via the following [3]:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \mathbf{C}_1(\phi) \left(\mathbf{C}_2(\theta) \left(\mathbf{C}_3(\psi) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} \right) + \begin{bmatrix} 0 \\ 0 \\ \dot{\phi} \end{bmatrix} \right). \quad (39)$$

We can now convert from $(\dot{\phi}, \dot{\theta}, \dot{\psi})$ to (p, q, r) , so to go the other direction, we can take the inverse and simplify. We end up with the following equations:

$$\dot{\phi} = p + r \cos \phi \tan \theta + q \sin \phi \tan \theta, \quad (40)$$

$$\dot{\theta} = q \cos \phi - r \sin \phi, \quad (41)$$

$$\dot{\psi} = r \cos \phi \sec \theta + q \sin \phi \sec \theta. \quad (42)$$

Eqs. (36 - 38) and Eqs. (40 - 42), along with Eqs. (28 - 33), form a full set of 12 ordinary differential equations, which form the basis for our state space.

III. State Space Model and LQR Control

A. Constructing a State Space

As described previously, \mathbf{x} , our state space, takes the form of a column vector,

$$\mathbf{x} = \begin{bmatrix} x & y & z & u & v & w & \phi & \theta & \psi & p & q & r \end{bmatrix}^T, \quad (43)$$

In addition, let \mathbf{u} represent the control inputs to our system, another column vector of the form,

$$\mathbf{u} = \begin{bmatrix} T & \tau_r & \xi & \zeta \end{bmatrix}^T. \quad (44)$$

Taken together, \mathbf{x} and \mathbf{u} may be used to develop a system of nonlinear equations,

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \quad (45)$$

equivalent to Eqs. (36 - 38), Eqs. (40 - 42), and Eqs. (28 - 33). However, to implement a LQR controller it is necessary to linearize Eq. (45) such that it may be written as,

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \quad (46)$$

where \mathbf{A} and \mathbf{B} represent the constant coefficients of \mathbf{x} and \mathbf{u} in the linearized system and are known as the state and input matrices respectively. Eq. (46) is known as the state space representation of Eq. (45) and may be created using a first-order Taylor series approximation of Eq. (45),

$$\mathbf{f}(\mathbf{x}, \mathbf{u}) \approx \mathbf{f}(\mathbf{x}_0, \mathbf{u}_0) + \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{(\mathbf{x}_0, \mathbf{u}_0)} (\mathbf{x} - \mathbf{x}_0) + \left. \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right|_{(\mathbf{x}_0, \mathbf{u}_0)} (\mathbf{u} - \mathbf{u}_0), \quad (47)$$

about the base point, $(\mathbf{x}_0, \mathbf{u}_0)$. Comparing Eq. (47) to Eq. (46) it may be seen that,

$$\mathbf{A} = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{(\mathbf{x}_0, \mathbf{u}_0)} (\mathbf{x}) ; \mathbf{B} = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right|_{(\mathbf{x}_0, \mathbf{u}_0)} (\mathbf{u}) . \quad (48)$$

In addition, Eq. (47) has a set of constant terms,

$$\mathbf{d} = \mathbf{f}(\mathbf{x}_0, \mathbf{u}_0) - \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{(\mathbf{x}_0, \mathbf{u}_0)} (\mathbf{x}_0) - \left. \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right|_{(\mathbf{x}_0, \mathbf{u}_0)} (\mathbf{u}_0) , \quad (49)$$

which is considered to be a disturbance not found in Eq. (46). \mathbf{d} represents a state-independent constant offset or an inhomogeneity and accordingly alters Eq. (46) to the form,

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{d} . \quad (50)$$

There are a variety of methods to resolve this affine nonlinearity, but for our purposes, we chose to counteract it through the addition of a feed-forward term to \mathbf{u} —the rationale being that a constant upwards thrust would best ensure stability against the constant downwards pull of gravity.

In a conventional state space, the system's equilibrium is defined to be the conditions under which,

$$\dot{\mathbf{x}} = \mathbf{0} . \quad (51)$$

In a traditional homogeneous state space, that would imply \mathbf{x} and \mathbf{u} are both $\mathbf{0}$. However, the presence of \mathbf{d} prevents Eq. (50) from reaching equilibrium, thus necessitating the aforementioned feed-forward control.

To introduce said feed-forward control, \mathbf{u}_{eq} , let

$$\delta \mathbf{u} = \mathbf{u} - \mathbf{u}_{eq} \implies \mathbf{u} = \delta \mathbf{u} + \mathbf{u}_{eq} . \quad (52)$$

be a change of variables given by Lloyd [4] such that,

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{d} = \mathbf{A}\mathbf{x} + \mathbf{B}(\delta \mathbf{u} + \mathbf{u}_{eq}) + \mathbf{d} . \quad (53)$$

By taking \mathbf{u}_{eq} to be constant along an episode of the gain scheduler and applying Eq. (51), the above reduces to,

$$\mathbf{B}\mathbf{u}_{eq} = -\mathbf{d} . \quad (54)$$

Eq. (54) is underdetermined because $\delta \mathbf{u} \in \mathbb{R}^4$ and $\mathbf{d} \in \mathbb{R}^{12}$, so it is impossible to solve analytically. To resolve this problem, we turn to the least-squares solution of an arbitrary affine system. Given a system,

$$\mathbf{A}\mathbf{x} = \mathbf{b} , \quad (55)$$

where $\mathbf{x} \in \mathbb{R}^N$ is an unknown column vector and $\mathbf{A} \in \mathbb{R}^{N \times N}$ and $\mathbf{b} \in \mathbb{R}^N$ have only constant components, its least-squares solution, \mathbf{x}_* , has the form

$$\mathbf{r} = \|\mathbf{A}\mathbf{x}_* - \mathbf{b}\| \leq \|\mathbf{A}\mathbf{x} - \mathbf{b}\| , \quad (56)$$

where $\|\mathbf{r}\|$ is the minimal residual of the system [5]. It can be shown that the Moore-Penrose pseudoinverse, \mathbf{A}^+ satisfies the property,

$$\mathbf{x}_* = \mathbf{A}^+ \mathbf{b} , \quad (57)$$

and thus can be used to directly compute the least-squares solution to an overdetermined system. However, it is important to note that the pseudoinverse does not always satisfy the definition of the traditional matrix inverse.

$$\mathbf{A}\mathbf{A}^+ \neq \mathbf{I} \quad \forall \mathbf{A} . \quad (58)$$

Equipped with the Moore-Penrose inverse, we can return to Eq. (54), and by applying Eq. (57), we obtain

$$\mathbf{u}_{eq} = -\mathbf{B}^+ \mathbf{d} . \quad (59)$$

Using Eq. (59), Eq. (50) may be rewritten as

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}(\delta \mathbf{u} - \mathbf{B}^+ \mathbf{d}) + \mathbf{d} . \quad (60)$$

It has been experimentally shown that for *Skipper* the components of \mathbf{r} are small enough that

$$\mathbf{B}\mathbf{B}^+ \mathbf{d} \approx \mathbf{d} , \quad (61)$$

Thus, we have, in effect, eliminated the affine nonlinearity from Eq. (46)*.

*When trimming for hovering the high degree of sparsity in \mathbf{B} and \mathbf{d} make this relation exact.

B. Negative Feedback and LQR Control

To introduce negative feedback to our system we can define,

$$\delta \mathbf{u} = -\mathbf{K}\mathbf{x} , \quad (62)$$

where \mathbf{K} is known as the gain matrix. The full control law for *Skipper*, including both feed-forward and negative feedback, is thus given by,

$$\mathbf{u} = -\mathbf{K}\mathbf{x} - \mathbf{B}^+ \mathbf{d} , \quad (63)$$

and may be introduced into Eq. (50) to yield,

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} - \mathbf{B}\mathbf{K}\mathbf{x} . \quad (64)$$

\mathbf{K} may now be computed using LQR control. To begin, let J , be a cost-functional for an arbitrary system of the form,

$$J = \int_0^{\infty} \mathbf{x}^T \mathbf{Q}\mathbf{x} + \delta \mathbf{u}^T \mathbf{R}\delta \mathbf{u} dt , \quad (65)$$

where \mathbf{Q} and \mathbf{R} are diagonal weight matrices applied to \mathbf{x} and \mathbf{u} respectively. We wish to find \mathbf{u}_{eq} (and thus \mathbf{K}) which minimizes J . This will occur when J is brought to zero as quickly as possible by driving \mathbf{x} and $\delta \mathbf{u}$ to zero because the quadratic form of J eliminates any notion of negative cost and is strictly convex. More precisely, we wish to find an optimal $\delta \mathbf{u}$, $\delta \mathbf{u}_*$, such that,

$$\delta \mathbf{u}_* = \arg \min_{\delta \mathbf{u}} J; \quad \delta \mathbf{u} = -\mathbf{K}\mathbf{x} . \quad (66)$$

It has been found that for a cost-functional of LQR form there exists an exact solution known to Eq. (66) known as the Continuous Algebraic Ricatti Equation (CARE) [6],

$$\mathbf{0} = \mathbf{S}\mathbf{A} + \mathbf{A}^T \mathbf{S} - \mathbf{S}\mathbf{B}\mathbf{R}^{-1} \mathbf{B}^T \mathbf{S} + \mathbf{Q} . \quad (67)$$

\mathbf{S} refers to the solution of the CARE, and can be used to directly compute \mathbf{K} using,

$$\mathbf{K} = \mathbf{R}^{-1} \mathbf{B}^T \mathbf{S} . \quad (68)$$

For our purposes, the MATLAB function `icare()` was used to compute \mathbf{K} numerically, because of its ability to solve CARE when \mathbf{A} is singular, an issue which occurs when trimming for hovering or near-hovering.

C. Automatic Gain Scheduling

Since the dynamics of the system are highly non-linear, the linearized system given by Eq. (46) and corresponding control law given in Eq. (63) rapidly become inaccurate as time progresses. To counteract this, on each guidance loop, the controller computes both the linear and nonlinear predictions for \mathbf{x} and compares them. If $|\mathbf{x}_{linear} - \mathbf{x}_{nonlinear}| > e$, where e is the norm of the component-wise tolerance for state deviation from the linear approximation, $\mathbf{x}_{nonlinear}$ and the current \mathbf{u} are chosen to be the new base-point about which the system is linearized. This, in turn, leads to new values for \mathbf{A} , \mathbf{B} , \mathbf{K} , and \mathbf{d} , which are fed back into the controller. Because this process is performed automatically it avoids the need for the lookup tables and pre-selected base points used in conventional pre-flight gain scheduling. It is important to note that even though we have the ability to compute an approximation for $\mathbf{x}_{nonlinear}$, we still must compute the linear approximation because we cannot get \mathbf{K} , an integral part of our control scheme, otherwise. A Simulink diagram for *Skipper*'s controller, including the gain scheduling, is shown in Figure 1.

IV. Results

For testing purposes, the following parameters were used:

$$\begin{aligned} m &= 0.01539 \text{ slug} , \\ g &= 32.2 \text{ ft/s}^2 , \\ H &= 0.458 \text{ ft} , \\ R &= 1.500 \text{ ft} , \\ \rho &= 0.750 \text{ ft} , \end{aligned}$$

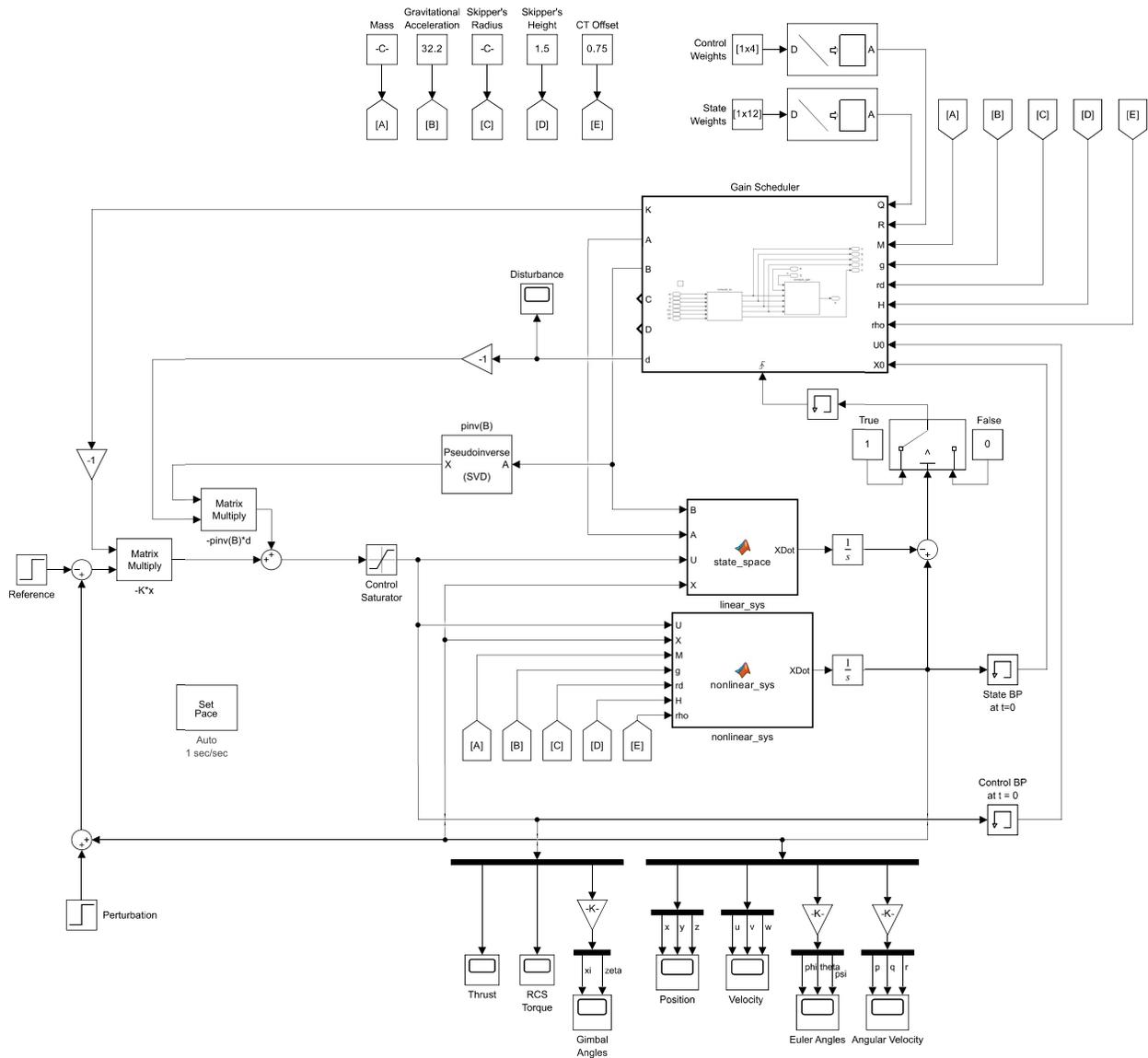


Fig. 1 Simulink model used to control and simulate the dynamics of *Skipper*.

where H and R represent the *Skipper*'s height and radius respectively. These were used to approximate J_{xx} , J_{yy} , and J_{zz} using the following formulas for the rotational inertia of a cylinder in a body-axis-fixed frame,

$$J_{xx} = \frac{mR^2}{2}, \quad (69)$$

$$J_{yy} = \frac{m(3R^2 + H^2)}{12} = J_{zz}. \quad (70)$$

In addition, a saturation block was added in cascade with the controller to restrict the operating range of the control inputs to:

$$\begin{aligned} 0 &\leq T \leq 2.2 \text{ lb}_f, \\ -1.157 &\leq \tau_r \leq 1.157 \text{ lb}_f\text{-ft}, \\ -10 &\leq \xi \leq 10 \text{ deg}, \\ -10 &\leq \zeta \leq 10 \text{ deg}. \end{aligned}$$

In addition, the following sets of LQR weights were used,

$$\mathbf{Q} = \text{diag}(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 10, 10) \quad (71)$$

$$\mathbf{R} = \text{diag}(1, 1, 10, 10). \quad (72)$$

The robustness of the control scheme was tested by testing *Skipper*'s ability to remain hovering at its initial position while under a constant y-horizontal wind at speeds ranging from 0 to 5 ft/s in increments of 0.1 m/s. The results of this analysis are shown in and Figure 3.

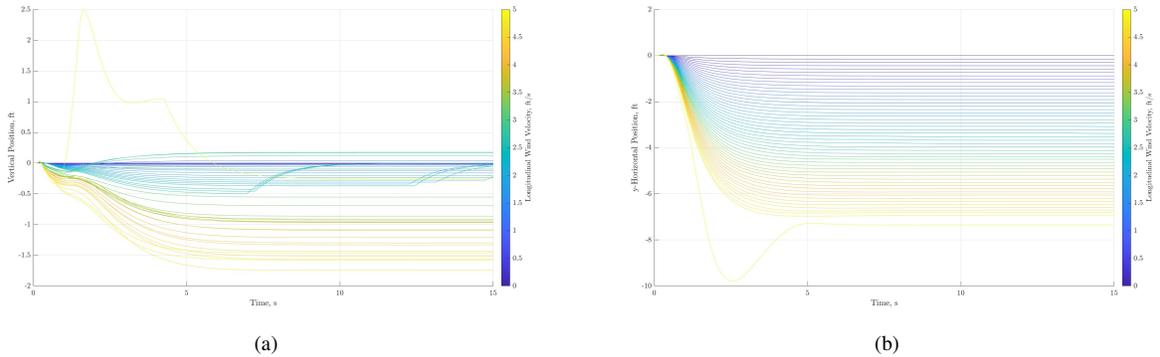
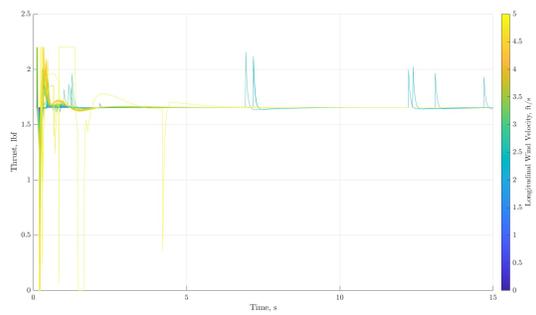


Fig. 2 Vertical (a) and y-horizontal (b) position under wind.

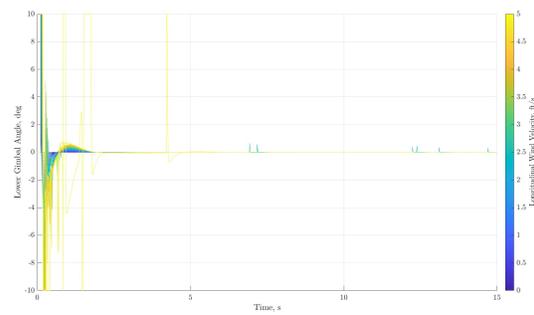
Skipper was able to maintain its stability while under the full range of wind speeds but not its equilibrium. As seen in Figure 2 at steady state there is a constant offset, likely due to the remaining affinity of the system after linearization as demonstrated by Eq. (56). The control response can be seen in Figure 3, and for the most part rapidly drives the system to its steady-state value. In addition, the feed-forward control term works as intended and the steady-state thrust counteracts *Skipper*'s weight as a result.

Then, *Skipper* was tasked with moving to a position \dagger of $(x, y, z) = (5, 5, 5)$ ft and then hovering upright at that spot indefinitely as seen in Figure 4. The controller makes use of its thrust vectoring ability to fly at an angle, also shown in Figure 4, although it struggles to maintain that angle over an extended period of time. This leads to gyroscopic behavior in both the angular rates and gimbal angle positions, seen in Figure 5 and Figure 6 respectively.

\dagger This can be accomplished through a change of variables similar to that of Eq. (52) via $\delta\mathbf{x} = \mathbf{x} - \mathbf{x}_{ref}$, where \mathbf{x}_{ref} is the target state [4].

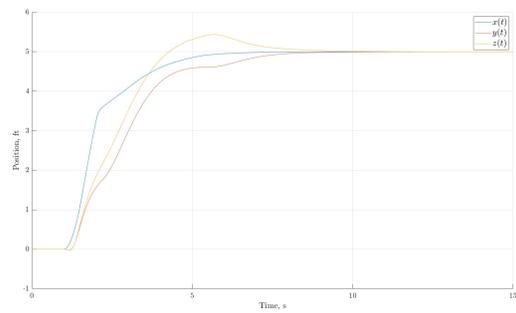


(a)

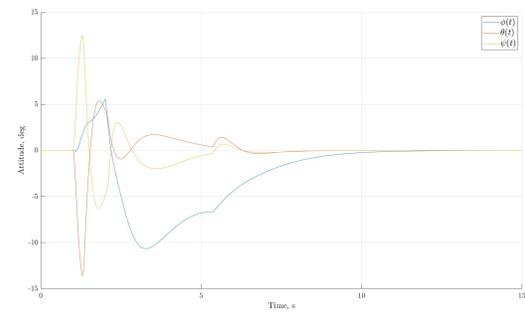


(b)

Fig. 3 Thrust (a) and lower gimbal angle (b) while under wind.

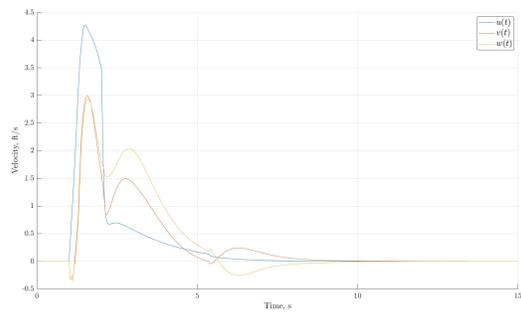


(a)

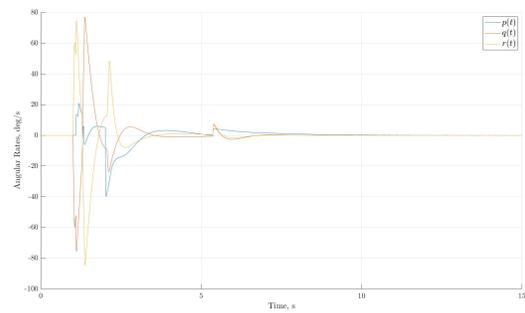


(b)

Fig. 4 Position (a) and attitude (b) while tracking a reference.



(a)



(b)

Fig. 5 Velocity (a) and angular rates (b) while tracking a reference.

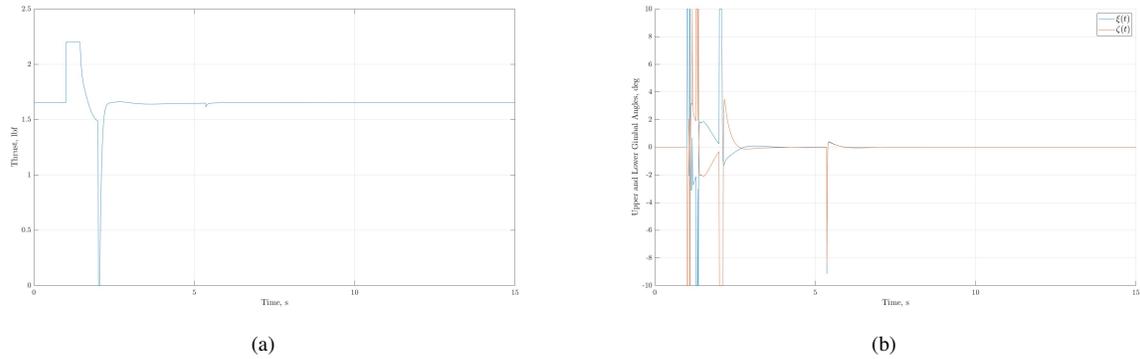


Fig. 6 Thrust (a) and gimbal angles (b) while tracking a reference.

V. Conclusion

In summary, we first derived the equations of motion governing the dynamics of a mono-propelled VTOL drone known as *Skipper*. We then discussed the linearization of this system and state-space representation, followed by the implementation of a LQR controller. We then implemented a Simulink model with reasonable modeling parameters for the control scheme. Finally, the ability of *Skipper* to maintain stability under a constant crosswind or while reference tracking was discussed.

Further research would likely focus on eliminating steady-state error via the implementation of a linear-quadratic-integrator controller. This would also remove the need for least-squares affinity reduction. A Kalman filter could also be added to improve stability in the face of noise. In addition, an optimal guidance law could be implemented to allow for *Skipper* to track an arbitrary reference otherwise far outside its immediate linear operating range, also improving its wind compensation. More rigorous methods of gain tuning, such as a machine learning approach, could also be considered. Finally, the control scheme detailed in this paper could be deployed on a real-world version of *Skipper* to demonstrate its practical efficacy.

References

- [1] dos Santos, P., and Oliveira, P., “Thrust Vector Control and State Estimation Architecture for Low-Cost Small-Scale Launchers,” *arXiv*, 2023. <https://doi.org/10.48550/arXiv.2303.16983>.
- [2] Ganbold, A., “Design of a Linear Quadratic Gaussian Control System for a Thrust Vector Controlled Rocket,” Master’s thesis, San Jose State University, 2023.
- [3] Cook, M. V., *Flight Dynamics Principles: A Linear Systems Approach to Aircraft Stability and Control*, Butterworth-Heinemann, 2012.
- [4] Lloyd, A., *Linearization of Differential Equation Models*, North Carolina State University, retrieved 2 March 2025. URL <http://alun.math.ncsu.edu/wp-content/uploads/sites/2/2017/01/linearization.pdf>.
- [5] MacAusland, R., *The Moore-Penrose Inverse and Least Squares*, Course Notes for University of Puget Sound MATH 420, retrieved 2 March 2025. URL <http://buzzard.ups.edu/courses/2014spring/420projects/math420-UPS-spring-2014-macausland-pseudo-inverse.pdf>.
- [6] Tedrake, R., *Underactuated Robotics: Algorithms for Walking, Running, Swimming, Flying, and Manipulation*, Course Notes for MIT 6.832, 2023. URL <https://underactuated.csail.mit.edu>.