Quadrotor Dynamics and Control

Randal W. Beard
Brigham Young University

February 19, 2008

1 Reference Frames

This section describes the various reference frames and coordinate systems that are used to describe the position of orientation of aircraft, and the transformation between these coordinate systems. It is necessary to use several different coordinate systems for the following reasons:

- Newton's equations of motion are given the coordinate frame attached to the quadrotor.
- Aerodynamics forces and torques are applied in the body frame.
- On-board sensors like accelerometers and rate gyros measure information with respect to the body frame. Alternatively, GPS measures position, ground speed, and course angle with respect to the inertial frame.
- Most mission requirements like loiter points and flight trajectories, are specified in the inertial frame. In addition, map information is also given in an inertial frame.

One coordinate frame is transformed into another through two basic operations: rotations and translations. Section 1.1 develops describes rotation matrices and their use in transforming between coordinate frames. Section 1.2 describes the specific coordinate frames used for micro air vehicle systems. In Section 1.3 we derive the Coriolis formula which is the basis for transformations between between between translating and rotating frames.

1.1 Rotation Matrices

We begin by considering the two coordinate systems shown in Figure 1. The

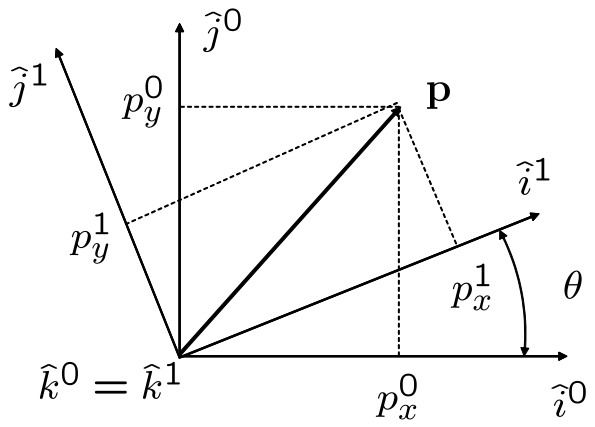

Figure 1: Rotation in 2D

vector $\mathbf{p}$ can be expressed in both the $\mathcal{F}^0$ frame (specified by $(\hat{i}^0, \hat{j}^0, \hat{k}^0)$) and in the $\mathcal{F}^1$ frame (specified by $(\hat{i}^1, \hat{j}^1, \hat{k}^1)$). In the $\mathcal{F}^0$ frame we have

$$\mathbf{p} = p_x^0 \hat{i}^0 + p_y^0 \hat{j}^0 + p_z^0 \hat{k}^0.$$

Alternatively in the $\mathcal{F}^1$ frame we have

$$\mathbf{p} = p_x^1 \hat{i}^1 + p_y^1 \hat{j}^1 + p_z^1 \hat{k}^1.$$

Setting these two expressions equal to each other gives

$$p_x^1 \hat{i}^1 + p_y^1 \hat{j}^1 + p_z^1 \hat{k}^1 = p_x^0 \hat{i}^0 + p_y^0 \hat{j}^0 + p_z^0 \hat{k}^0.$$

Taking the dot product of both sides with $\hat{i}^1$, $\hat{j}^1$, and $\hat{k}^1$ respectively, and stacking the result into matrix form gives

$$\mathbf{p}^1 \triangleq \begin{pmatrix} p_x^1 \\ p_y^1 \\ p_z^1 \end{pmatrix} = \begin{pmatrix} \hat{i}^1 \cdot \hat{i}^0 & \hat{i}^1 \cdot \hat{j}^0 & \hat{i}^1 \cdot \hat{k}^0 \\ \hat{j}^1 \cdot \hat{i}^0 & \hat{j}^1 \cdot \hat{j}^0 & \hat{j}^1 \cdot \hat{k}^0 \\ \hat{k}^1 \cdot \hat{i}^0 & \hat{k}^1 \cdot \hat{j}^0 & \hat{k}^1 \cdot \hat{k}^0 \end{pmatrix} \begin{pmatrix} p_x^0 \\ p_y^0 \\ p_z^0 \end{pmatrix}.$$

From the geometry of Figure 1 we get

$$\mathbf{p}^1 = R_0^1 \mathbf{p}^0, \tag{1}$$

where

$$R_0^1 \triangleq \begin{pmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

The notation R_0^1 is used to denote a rotation matrix from coordinate frame $\mathcal{F}^0$ to coordinate frame $\mathcal{F}^1$.

Proceeding in a similar way, a right-handed rotation of the coordinate system about the y-axis gives

$$R_0^1 \triangleq \begin{pmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{pmatrix},$$

and a right-handed rotation of the coordinate system about the x-axis resultes in

$$R_0^1 \triangleq \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) \\ 0 & -\sin(\theta) & \cos(\theta) \end{pmatrix}.$$

As pointed out in [1], the negative sign on the sin term appears above the line with only ones and zeros.

The matrix R_0^1 in the above equations are examples of a more general class of rotation matrices that have the following properties:

P.1. $(R_a^b)^{-1} = (R_a^b)^T = R_b^a$.

P.2. $R_b^c R_a^b = R_a^c$.

P.3. $\det R_a^b = 1$.

In the derivation of Equation (1) note that the vector $\mathbf{p}$ remains constant and the new coordinate frame $\mathcal{F}^1$ was obtained by rotating $\mathcal{F}^0$ through a *righted-handed* rotation of angle θ.

We will now derive a formula, called the *rotation formula* that performs a *left-handed* rotation of a vector $\mathbf{p}$ about another vector $\hat{n}$ by an angle of μ. Our derivation follows that given in [1]. Consider Figure 2 which is similar to Figure 1.2-2 in [1]. The vector $\mathbf{p}$ is rotated, in a left-handed sense, about a unit vector $\hat{n}$ by an angle of μ to produce the vector $\mathbf{q}$. The angle between $\mathbf{p}$ and $\hat{n}$ is ϕ. By geometry we have that

$$\mathbf{q} = \vec{ON} + \vec{NW} + \vec{WQ}. \tag{2}$$

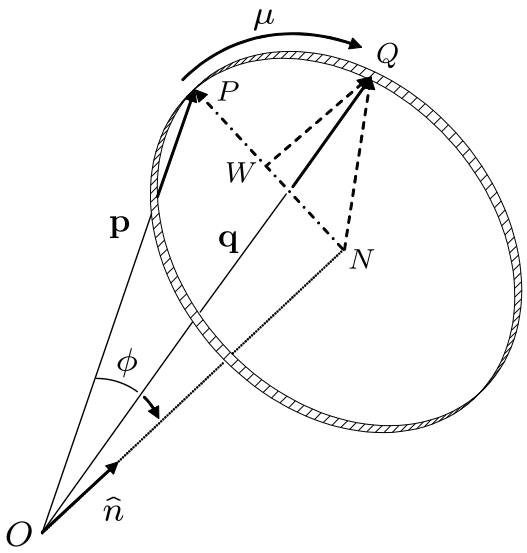

Figure 2: Left-handed rotation of a vector $\mathbf{p}$ about the unit vector $\hat{n}$ by an angle of μ to obtain the vector $\mathbf{q}$.

The vector $\vec{ON}$ can be found by taking the projection of $\mathbf{p}$ on the unit vector $\hat{n}$ in the direction of $\hat{n}$:

$$\vec{ON} = (\mathbf{p} \cdot \hat{n})\,\hat{n}.$$

The vector $\vec{NW}$ is in the direction of $\mathbf{p} - \vec{ON}$ with a length of $NQ\cos\mu$. Noting that the length NQ equals the length NP which is equal to $\left\|\mathbf{p} - \vec{ON}\right\|$ we get that

$$\begin{aligned}\vec{NW} &= \frac{\mathbf{p} - (\mathbf{p}\cdot\hat{n})\,\hat{n}}{\|\mathbf{p} - (\mathbf{p}\cdot\hat{n})\,\hat{n}\|} NQ\cos\mu \\ &= (\mathbf{p} - (\mathbf{p}\cdot\hat{n})\,\hat{n})\cos\mu.\end{aligned}$$

The vector $\vec{WQ}$ is perpendicular to both $\mathbf{p}$ and $\hat{n}$ and has length $NQ\sin\mu$. Noting that $NQ = \|\mathbf{p}\|\sin\phi$ we get

$$\begin{aligned}\vec{WQ} &= \frac{\mathbf{p}\times\hat{n}}{\|\mathbf{p}\|\sin\phi} NQ\sin\mu \\ &= -\hat{n}\times\mathbf{p}\sin\mu.\end{aligned}$$

Therefore Equation (2) becomes

$$\mathbf{q} = (1-\cos\mu)\,(\mathbf{p}\cdot\hat{n})\,\hat{n} + \cos\mu\mathbf{p} - \sin\mu\,(\hat{n}\times\mathbf{p})\,, \tag{3}$$

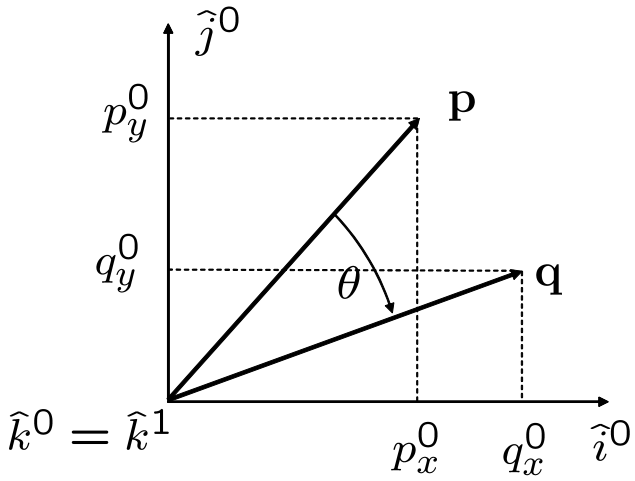

Figure 3: Rotation of $\mathbf{p}$ about the z-axis.

which is called the rotation formula.

As an example of the application of Equation (3) consider a left handed rotation of a vector $\mathbf{p}^0$ in frame $\mathcal{F}^0$ about the z-axis as shown in Figure 3.

Using the rotation formula we get

$$\begin{aligned}
\mathbf{q}^0 &= (1-\cos\theta)(\mathbf{p}\cdot\hat{n})\hat{n} + \cos\phi\mathbf{p} - \sin\phi\hat{n}\times\mathbf{p} \\
&= (1-\cos\phi)p_z^0\begin{pmatrix}0\\0\\1\end{pmatrix} + \cos\phi\begin{pmatrix}p_x^0\\p_y^0\\p_z^0\end{pmatrix} - \sin\phi\begin{pmatrix}-p_y^0\\p_x^0\\0\end{pmatrix} \\
&= \begin{pmatrix}\cos\phi & \sin\phi & 0\\ -\sin\phi & \cos\phi & 0\\ 0 & 0 & 1\end{pmatrix}\mathbf{p}^0 \\
&= R_0^1\mathbf{p}^0.
\end{aligned}$$

Note that the rotation matrix R_0^1 can be interpreted in two different ways. The first interpretation is that it transforms the fixed vector $\mathbf{p}$ from an expression in frame $\mathcal{F}^0$ to an expression in frame $\mathcal{F}^1$ where $\mathcal{F}^1$ has been obtained from $\mathcal{F}^0$ by a right-handed rotation. The second interpretation is that it rotates a vector $\mathbf{p}$ though a left-handed rotation to a new vector $\mathbf{q}$ in the same reference frame. Right-handed rotations of vectors are obtained by using $(R_0^1)^T$.

1.2 Quadrotor Coordinate Frames

For quadrotors there are several coordinate systems that are of interest. In this section we will define and describe the following coordinate frames: the inertial frame, the vehicle frame, the vehicle-1 frame, the vehicle-2 frame, and the body frame. Throughout the book we assume a flat, non-rotating earth: a valid assumption for quadrotors.

1.2.1 The inertial frame $\mathcal{F}^i$.

The inertial coordinate system is an earth fixed coordinate system with origin at the defined home location. As shown in Figure 4, the unit vector $\hat{i}^i$ is directed North, $\hat{j}^i$ is directed East, and $\hat{k}^i$ is directed toward the center of the earth.

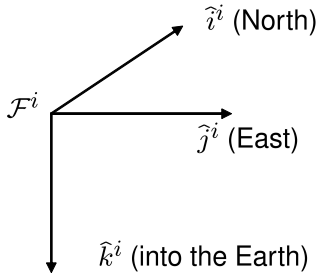

Figure 4: The inertial coordinate frame. The x-axis points North, the y-axis points East, and the z-axis points into the Earth.

1.2.2 The vehicle frame $\mathcal{F}^v$.

The origin of the vehicle frame is at the center of mass of the quadrotor. However, the axes of $\mathcal{F}^v$ are aligned with the axis of the inertial frame $\mathcal{F}^i$. In other words, the unit vector $\hat{i}^v$ points North, $\hat{j}^v$ points East, and $\hat{k}^v$ points toward the center of the earth, as shown in Figure 5.

1.2.3 The vehicle-1 frame $\mathcal{F}^{v1}$.

The origin of the vehicle-1 frame is identical to the vehicle frame, i.e, the the center of gravity. However, $\mathcal{F}^{v1}$ is positively rotated about $\hat{k}^v$ by the yaw angle ψ so that if the airframe is not rolling or pitching, then $\hat{i}^{v1}$ would point out the nose

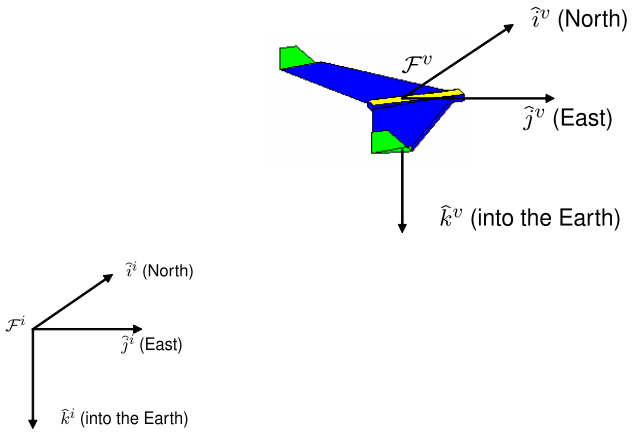

Figure 5: The vehicle coordinate frame. The x-axis points North, the y-axis points East, and the z-axis points into the Earth.

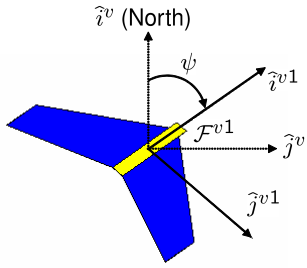

Figure 6: The vehicle-1 frame. If the roll and pitch angels are zero, then the x-axis points out the nose of the airframe, the y-axis points out the right wing, and the z-axis points into the Earth.

of the airframe, $\hat{j}^{v1}$ points out the right wing, and $\hat{k}^{v1}$ is aligned with $\hat{k}^v$ and points into the earth. The vehicle-1 frame is shown in Figure 6.

The transformation from $\mathcal{F}^v$ to $\mathcal{F}^{v1}$ is given by

$$p^{v1} = R_v^{v1}(\psi)p^v,$$

where

$$R_v^{v1}(\psi) = \begin{pmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

1.2.4 The vehicle-2 frame $\mathcal{F}^{v2}$.

The origin of the vehicle-2 frame is again the center of gravity and is obtained by rotating the vehicle-1 frame in a right-handed rotation about the $\hat{j}^{v1}$ axis by the pitch angle θ. If the roll angle is zero, then $\hat{i}^{v2}$ points out the nose of the airframe, $\hat{j}^{v2}$ points out the right wing, and $\hat{k}^{v2}$ points out the belly, a shown in Figure 7.

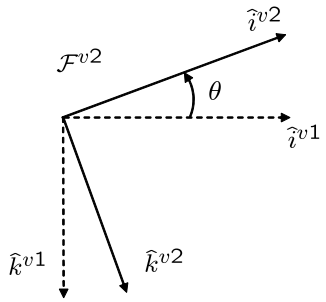

Figure 7: The vehicle-2 frame. If the roll angel is zero, then the x-axis points out the nose of the airframe, the y-axis points out the right wing, and the z-axis points out the belly.

The transformation from $\mathcal{F}^{v1}$ to $\mathcal{F}^{v2}$ is given by

$$p^{v2} = R^{v2}_{v1}(\theta) p^{v1},$$

where

$$R^{v2}_{v1}(\theta) = \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix}.$$

1.2.5 The body frame $\mathcal{F}^{b}$.

The body frame is obtained by rotating the vehicle-2 frame in a right handed rotation about $\hat{i}^{v2}$ by the roll angle ϕ. Therefore, the origin is the center-of-gravity, $\hat{i}^{b}$ points out the nose of the airframe, $\hat{j}^{b}$ points out the right wing, and $\hat{k}^{b}$ points out the belly. The body frame is shown in Figure 8.

The transformation from $\mathcal{F}^{v2}$ to $\mathcal{F}^{b}$ is given by

$$p^{b} = R^{b}_{v2}(\phi) p^{v2},$$

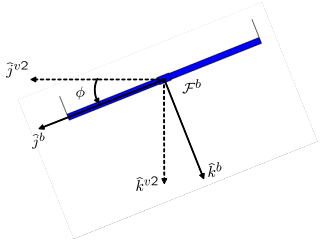

Figure 8: The body frame. The x-axis points out the nose of the airframe, the y-axis points out the right wing, and the z-axis points out the belly.

where

$$R_{v2}^b(\phi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{pmatrix}.$$

The transformation from the vehicle frame to the body frame is given by

$$\begin{aligned} R_v^b(\phi,\theta,\psi) &= R_{v2}^b(\phi) R_{v1}^{v2}(\theta) R_v^{v1}(\psi) \\ &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{pmatrix} \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix} \begin{pmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} c\theta c\psi & c\theta s\psi & -s\theta \\ s\phi s\theta c\psi - c\phi s\psi & s\phi s\theta s\psi + c\phi c\psi & s\phi c\theta \\ c\phi s\theta c\psi + s\phi s\psi & c\phi s\theta s\psi - s\phi c\psi & c\phi c\theta \end{pmatrix}, \end{aligned}$$

where $c\phi \triangleq \cos\phi$ and $s\phi \triangleq \sin\phi$.

1.3 Equation of Coriolis

In this section we provide a simple derivation of the famous equation of Coriolis. We will again follow the derivation given in [1].

Suppose that we are given two coordinate frames $\mathcal{F}^i$ and $\mathcal{F}^b$ as shown in Figure 9. For example, $\mathcal{F}^i$ might represent the inertial frame and $\mathcal{F}^b$ might represent the body frame of a quadrotor. Suppose that the vector $\mathbf{p}$ is moving in $\mathcal{F}^b$ and that $\mathcal{F}^b$ is rotating and translating with respect to $\mathcal{F}^i$. Our objective is to find the time derivative of $\mathbf{p}$ as seen from frame $\mathcal{F}^i$.

We will derive the appropriate equation through two steps. Assume first that $\mathcal{F}^b$ is not rotating with respect to $\mathcal{F}^i$. Denoting the time derivative of $\mathbf{p}$ in frame

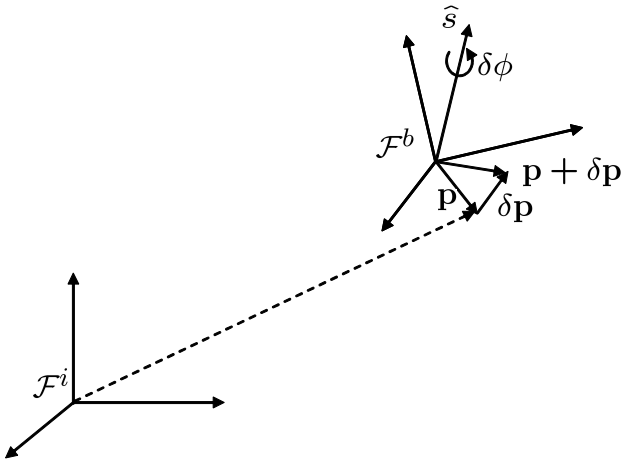

Figure 9: Derivation of the equation of Coriolis.

$\mathcal{F}^i$ as $\frac{d}{dt_i}\mathbf{p}$ we get

$$\frac{d}{dt_i}\mathbf{p} = \frac{d}{dt_b}\mathbf{p}. \tag{4}$$

On the other hand, assume that $\mathbf{p}$ is fixed in $\mathcal{F}^b$ but that $\mathcal{F}^b$ is rotating with respect to $\mathcal{F}^i$, and let $\hat{s}$ be the instantaneous axis of rotation and $\delta\phi$ the (right-handed) rotation angle. Then the rotation formula (3) gives

$$\mathbf{p} + \delta\mathbf{p} = (1 - \cos(-\delta\phi))\hat{s}(\hat{s} \cdot \mathbf{p}) + \cos(-\delta\phi)\mathbf{p} - \sin(-\delta\phi)\hat{s} \times \mathbf{p}.$$

Using a small angle approximation and dividing both sides by δt gives

$$\frac{\delta\mathbf{p}}{\delta t} \approx \frac{\delta\phi}{\delta t}\hat{s} \times \mathbf{p}.$$

Taking the limit as $\delta t \to 0$ and defining the angular velocity of $\mathcal{F}^b$ with respect to $\mathcal{F}^i$ as $\omega_{b/i} \triangleq \hat{s}\dot{\phi}$ we get

$$\frac{d}{dt_i}\mathbf{p} = \omega_{b/i} \times \mathbf{p}. \tag{5}$$

Since differentiation is a linear operator we can combine Equations (4) and (5) to obtain

$$\frac{d}{dt_i}\mathbf{p} = \frac{d}{dt_b}\mathbf{p} + \omega_{b/i} \times \mathbf{p}, \tag{6}$$

which is the equation of Coriolis.

2 Kinematics and Dynamics

In this chapter we derive the expressions for the kinematics and the dynamics of a rigid body. While the expressions derived in this chapter are general to any rigid body, we will use notation and coordinate frames that are typical in the aeronautics literature. In particular, in Section 2.1 we define the notation that will be used for the state variables of a quadrotor. In Section 2.2 we derive the expressions for the kinematics, and in Section 2.3 we derive the dynamics.

2.1 Quadrotor State Variables

The state variables of the quadrotor are the following twelve quantities

$$
\begin{aligned}
p_n &= \text{the inertial (north) position of the quadrotor along } \hat{i}^i \text{ in } \mathcal{F}^i,\\
p_e &= \text{the inertial (east) position of the quadrotor along } \hat{j}^i \text{ in } \mathcal{F}^i,\\
h &= \text{the altitude of the aircraft measured along } -\hat{k}^i \text{ in } \mathcal{F}^i,\\
u &= \text{the body frame velocity measured along } \hat{i}^b \text{ in } \mathcal{F}^b,\\
v &= \text{the body frame velocity measured along } \hat{j}^b \text{ in } \mathcal{F}^b,\\
w &= \text{the body frame velocity measured along } \hat{k}^b \text{ in } \mathcal{F}^b,\\
\phi &= \text{the roll angle defined with respect to } \mathcal{F}^{v2},\\
\theta &= \text{the pitch angle defined with respect to } \mathcal{F}^{v1},\\
\psi &= \text{the yaw angle defined with respect to } \mathcal{F}^{v},\\
p &= \text{the roll rate measured along } \hat{i}^b \text{ in } \mathcal{F}^b,\\
q &= \text{the pitch rate measured along } \hat{j}^b \text{ in } \mathcal{F}^b,\\
r &= \text{the yaw rate measured along } \hat{k}^b \text{ in } \mathcal{F}^b.
\end{aligned}
$$

The state variables are shown schematically in Figure 10. The position (p_n, p_e, h) of the quadrotor is given in the inertial frame, with positive h defined along the negative Z axis in the inertial frame. The velocity (u, v, w) and the angular velocity (p, q, r) of the quadrotor are given with respect to the body frame. The Euler angles (roll ϕ, pitch θ, and yaw χ) are given with respect to the vehicle 2-frame, the vehicle 1-frame, and the vehicle frame respectively.

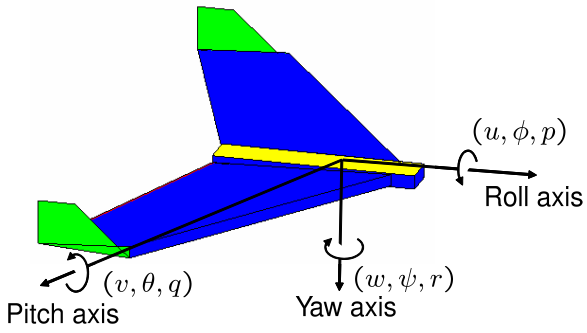

Figure 10: Definition of Axes

2.2 Quadrotor Kinematics

The state variables p_n, p_e, and $-h$ are inertial frame quantities, whereas the velocities u, v, and w are body frame quantities. Therefore the relationship between position and velocities is given by

$$\begin{aligned}\frac{d}{dt}\begin{pmatrix} p_n \\ p_e \\ -h \end{pmatrix} &= R_b^v \begin{pmatrix} u \\ v \\ w \end{pmatrix} \\ &= (R_v^b)^T \begin{pmatrix} u \\ v \\ w \end{pmatrix} \\ &= \begin{pmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{pmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix}.\end{aligned}$$

The relationship between absolute angles ϕ, θ, and ψ, and the angular rates p, q, and r is also complicated by the fact that these quantities are defined in different coordinate frames. The angular rates are defined in the body frame $\mathcal{F}^b$, whereas the roll angle ϕ is defined in $\mathcal{F}^{v2}$, the pitch angle θ is defined in $\mathcal{F}_{v1}$, and the yaw angle ψ is defined in the vehicle frame $\mathcal{F}^v$.

We need to relate p, q, and r to $\dot{\phi}$, $\dot{\theta}$, and $\dot{\psi}$. Since $\dot{\phi}$, $\dot{\theta}$, $\dot{\psi}$ are small and noting that

$$R_{v2}^b(\dot{\phi}) = R_{v1}^{v2}(\dot{\theta}) = R_v^{v1}(\dot{\psi}) = I,$$

we get

$$\begin{aligned}
\begin{pmatrix} p \\ q \\ r \end{pmatrix} &= R^b_{v2}(\dot{\phi}) \begin{pmatrix} \dot{\phi} \\ 0 \\ 0 \end{pmatrix} + R^b_{v2}(\phi) R^{v2}_{v1}(\dot{\theta}) \begin{pmatrix} 0 \\ \dot{\theta} \\ 0 \end{pmatrix} + R^b_{v2}(\phi) R^{v2}_{v1}(\theta) R_{v \to v1}(\dot{\psi}) \begin{pmatrix} 0 \\ 0 \\ \dot{\psi} \end{pmatrix} \\
&= \begin{pmatrix} \dot{\phi} \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{pmatrix} \begin{pmatrix} 0 \\ \dot{\theta} \\ 0 \end{pmatrix} + \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{pmatrix} \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ \dot{\psi} \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 & -s\theta \\ 0 & c\phi & s\phi c\theta \\ 0 & -s\phi & c\phi c\theta \end{pmatrix} \begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix}. \qquad (7)
\end{aligned}$$

Inverting we get

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)\sec(\theta) & \cos(\phi)\sec(\theta) \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix}. \qquad (8)$$

2.3 Rigid Body Dynamics

Let $\mathbf{v}$ be the velocity vector of the quadrotor. Newton's laws only hold in inertial frames, therefore Newton's law applied to the translational motion is

$$m\frac{d\mathbf{v}}{dt_i} = \mathbf{f},$$

where m is the mass of the quadrotor, $\mathbf{f}$ is the total applied to the quadrotor, and $\frac{d}{dt_i}$ is the time derivative in the inertial frame. From the equation of Coriolis we have

$$m\frac{d\mathbf{v}}{dt_i} = m\left(\frac{d\mathbf{v}}{dt_b} + \boldsymbol{\omega}_{b/i} \times \mathbf{v}\right) = \mathbf{f}, \qquad (9)$$

where $\boldsymbol{\omega}_{b/i}$ is the angular velocity of the airframe with respect to the inertial frame. Since the control force is computed and applied in the body coordinate system, and since $\boldsymbol{\omega}$ is measured in body coordinates, we will express Eq (9) in body coordinates, where $\mathbf{v}^b \stackrel{\triangle}{=} (u, v, w)^T$, and $\omega^b_{b/i} \stackrel{\triangle}{=} (p, q, r)^T$. Therefore, in body coordinates, Eq. (9) becomes

$$\begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} = \begin{pmatrix} rv - qw \\ pw - ru \\ qu - pv \end{pmatrix} + \frac{1}{m}\begin{pmatrix} f_x \\ f_y \\ f_z \end{pmatrix}, \qquad (10)$$

where $\mathbf{f}^b \triangleq (f_x, f_y, f_z)^T$.

For rotational motion, Newton's second law states that

$$\frac{d\mathbf{h}^b}{dt_i} = \mathbf{m},$$

where $\mathbf{h}$ is the angular momentum and $\mathbf{m}$ is the applied torque. Using the equation of Coriolis we have

$$\frac{d\mathbf{h}}{dt_i} = \frac{d\mathbf{h}}{dt_b} + \boldsymbol{\omega}_{b/i} \times \mathbf{h} = \mathbf{m}. \tag{11}$$

Again, Eq. (11) is most easily resolved in body coordinates where $\mathbf{h}^b = \mathbf{J}\boldsymbol{\omega}^{\boldsymbol{b}}_{\boldsymbol{b/i}}$ where $\mathbf{J}$ is the constant inertia matrix given by

$$\begin{aligned}\mathbf{J} &= \begin{pmatrix} \int (y^2+z^2)\,dm & -\int xy\,dm & -\int xz\,dm \\ -\int xy\,dm & \int (x^2+z^2)\,dm & -\int yz\,dm \\ -\int xz\,dm & -\int yz\,dm & \int (x^2+y^2)\,dm \end{pmatrix} \\ &\triangleq \begin{pmatrix} J_x & -J_{xy} & -J_{xz} \\ -J_{xy} & J_y & -J_{yz} \\ -J_{xz} & -J_{yz} & J_z \end{pmatrix}.\end{aligned}$$

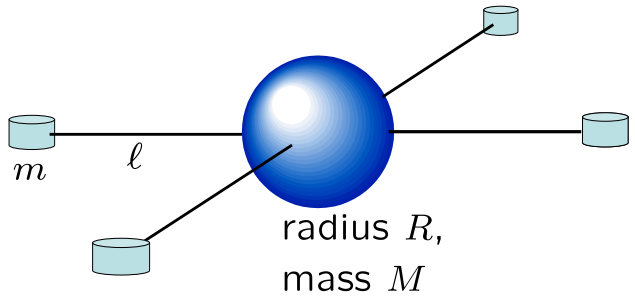

Figure 11: The moments of inertia for the quadrotor are calculated assuming a spherical dense center with mass M and radius R, and point masses of mass m located at a distance of ℓ from the center.

As shown in Figure 11, the quadrotor is essentially symmetric about all three axes, therefore $J_{xy} = J_{xz} = J_{yz} = 0$ which implies that

$$\mathbf{J} = \begin{pmatrix} J_x & 0 & 0 \\ 0 & J_y & 0 \\ 0 & 0 & J_z \end{pmatrix}.$$

Therefore

$$\mathbf{J}^{-1} = \begin{pmatrix} \frac{1}{J_x} & 0 & 0 \\ 0 & \frac{1}{J_y} & 0 \\ 0 & 0 & \frac{1}{J_z} \end{pmatrix}.$$

The inertia for a solid sphere is given by $J = 2MR^2/5$[2]. Therefore

$$\begin{aligned} J_x &= \frac{2MR^2}{5} + 2\ell^2 m \\ J_y &= \frac{2MR^2}{5} + 2\ell^2 m \\ J_z &= \frac{2MR^2}{5} + 4\ell^2 m. \end{aligned}$$

Defining $\mathbf{m}^b \triangleq (\tau_\phi, \tau_\theta, \tau_\psi)^T$ we can write Eq. (11) in body coordinates as

$$\begin{aligned} \begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} &= \begin{pmatrix} \frac{1}{J_x} & 0 & 0 \\ 0 & \frac{1}{J_y} & 0 \\ 0 & 0 & \frac{1}{J_z} \end{pmatrix} \left[\begin{pmatrix} 0 & r & -q \\ -r & 0 & p \\ q & -p & 0 \end{pmatrix} \begin{pmatrix} J_x & 0 & 0 \\ 0 & J_y & 0 \\ 0 & 0 & J_z \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix} + \begin{pmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{pmatrix} \right] \\ &= \begin{pmatrix} \frac{J_y - J_z}{J_x} qr \\ \frac{J_z - J_x}{J_y} pr \\ \frac{J_x - J_y}{J_z} pq \end{pmatrix} + \begin{pmatrix} \frac{1}{J_x}\tau_\phi \\ \frac{1}{J_y}\tau_\theta \\ \frac{1}{J_z}\tau_\psi \end{pmatrix}. \end{aligned}$$

The six degree of freedom model for the quadrotor kinematics and dynamics can be summarized as follows:

$$\begin{pmatrix} \dot{p}_n \\ \dot{p}_e \\ \dot{h} \end{pmatrix} = \begin{pmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ s\theta & -s\phi c\theta & -c\phi c\theta \end{pmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix} \tag{12}$$

$$\begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} = \begin{pmatrix} rv - qw \\ pw - ru \\ qu - pv \end{pmatrix} + \frac{1}{m} \begin{pmatrix} f_x \\ f_y \\ f_z \end{pmatrix}, \tag{13}$$

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & \sin\phi \tan\theta & \cos\phi \tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix} \tag{14}$$

$$\begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} = \begin{pmatrix} \frac{J_y - J_z}{J_x} qr \\ \frac{J_z - J_x}{J_y} pr \\ \frac{J_x - J_y}{J_z} pq \end{pmatrix} + \begin{pmatrix} \frac{1}{J_x}\tau_\phi \\ \frac{1}{J_y}\tau_\theta \\ \frac{1}{J_z}\tau_\psi \end{pmatrix}. \tag{15}$$

3 Forces and Moments

The objective of this section is to describe the forces and torques that act on the quadrotor. Since there are no aerodynamic lifting surfaces, we will assume that the aerodynamic forces and moments are negligible.

The forces and moments are primarily due to gravity and the four propellers.

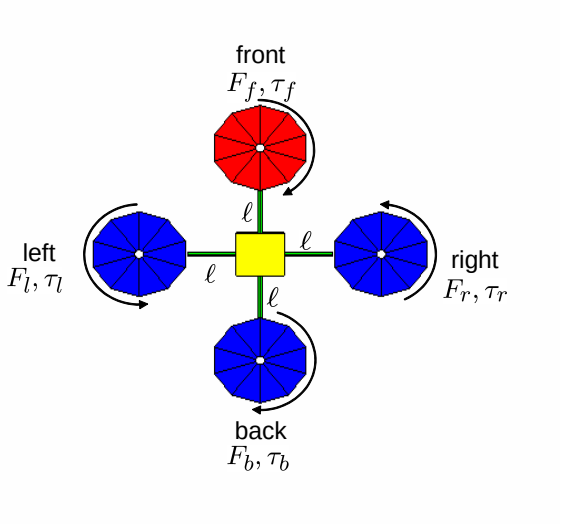

Figure 12: The top view of the quadrotor. Each motor produces an upward force F and a torque τ. The front and back motors spin clockwise and the right and left motors spin counterclockwise.

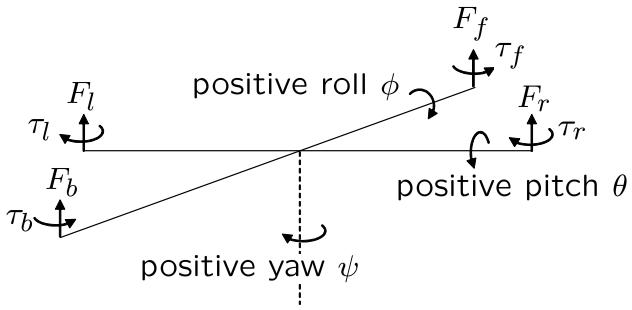

Figure 13: Definition of the forces and torques acting on the quadrotor.

Figure 12 shows a top view of the quadrotor systems. As shown in Figure 13, each motor produces a force F and a torque τ. The total force acting on the

quadrotor is given by

$$F = F_f + F_r + F_b + F_l.$$

The rolling torque is produced by the forces of the right and left motors as

$$\tau_\phi = \ell(F_l - F_r).$$

Similarly, the pitching torque is produced by the forces of the front and back motors as

$$\tau_\theta = \ell(F_f - F_b).$$

Due to Newton's third law, the drag of the propellers produces a yawing torque on the body of the quadrotor. The direction of the torque will be in the oppositive direction of the motion of the propeller. Therefore the total yawing torque is given by

$$\tau_\psi = \tau_r + \tau_l - \tau_f - \tau_b.$$

The lift and drag produced by the propellers is proportional to the square of the angular velocity. We will assume that the angular velocity is directly proportional to the pulse width modulation commend sent to the motor. Therefore, the force and torque of each motor can be expressed as

$$\begin{aligned} F_* &= k_1 \delta_* \\ \tau_* &= k_2 \delta_*, \end{aligned}$$

where k_1 and k_2 are constants that need to be determined experimentally, δ_* is the motor command signal, and $*$ represents f, r, b, and l.

Therefore, the forces and torques on the quadrotor can be written in matrix form as

$$\begin{pmatrix} F \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{pmatrix} = \begin{pmatrix} k_1 & k_1 & k_1 & k_1 \\ 0 & -\ell k_1 & 0 & \ell k_1 \\ \ell k_1 & 0 & \ell k_1 & 0 \\ -k_2 & k_2 & -k_2 & k_2 \end{pmatrix} \begin{pmatrix} \delta_f \\ \delta_r \\ \delta_b \\ \delta_l \end{pmatrix} \triangleq \mathcal{M} \begin{pmatrix} \delta_f \\ \delta_r \\ \delta_b \\ \delta_l \end{pmatrix}.$$

The control strategies derived in subsequent sections will specify forces and torques. The actual motors commands can be found as

$$\begin{pmatrix} \delta_f \\ \delta_r \\ \delta_b \\ \delta_l \end{pmatrix} = \mathcal{M}^{-1} \begin{pmatrix} F \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{pmatrix}.$$

Note that the pulse width modulation commands are required to be between zero and one.

In addition to the force exerted by the motor, gravity also exerts a force on the quadrotor. In the vehicle frame $\mathcal{F}^v$, the gravity force acting on the center of mass is given by

$$\mathbf{f}_g^v = \begin{pmatrix} 0 \\ 0 \\ mg \end{pmatrix}.$$

However, since $\mathbf{v}$ in Equation (13) is expressed in $\mathcal{F}^b$, we must transform to the body frame to give

$$\begin{aligned} \mathbf{f}_g^b &= R_v^b \begin{pmatrix} 0 \\ 0 \\ mg \end{pmatrix} \\ &= \begin{pmatrix} -mg\sin\theta \\ mg\cos\theta\sin\phi \\ mg\cos\theta\cos\phi \end{pmatrix}. \end{aligned}$$

Therefore, equations (12)–(15) become

$$\begin{pmatrix} \dot{p}_n \\ \dot{p}_e \\ \dot{h} \end{pmatrix} = \begin{pmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ s\theta & -s\phi c\theta & -c\phi c\theta \end{pmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix} \tag{16}$$

$$\begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} = \begin{pmatrix} rv - qw \\ pw - ru \\ qu - pv \end{pmatrix} + \begin{pmatrix} -g\sin\theta \\ g\cos\theta\sin\phi \\ g\cos\theta\cos\phi \end{pmatrix} + \frac{1}{m}\begin{pmatrix} 0 \\ 0 \\ -F \end{pmatrix}, \tag{17}$$

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix}, \tag{18}$$

$$\begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} = \begin{pmatrix} \frac{J_y - J_z}{J_x} qr \\ \frac{J_z - J_x}{J_y} pr \\ \frac{J_x - J_y}{J_z} pq \end{pmatrix} + \begin{pmatrix} \frac{1}{J_x}\tau_\phi \\ \frac{1}{J_y}\tau_\theta \\ \frac{1}{J_z}\tau_\psi \end{pmatrix}. \tag{19}$$

4 Simplified Models

Equations (16)–(19) are the equations of motion to be used in our six degree-of-freedom simulator. However, they are not appropriate for control design for

several reasons. The first reason is that they are too complicated to gain significant insight into the motion. The second reason is that the position and orientation are relative to the inertial world fixed frame, whereas camera measurements will measure position and orientation of the target with respect to the camera frame.

4.1 Model for estimation

For the quadrotor, we are not able to estimate the inertial position or the heading angle ψ. Rather, we will be interested in the relative position and heading of the quadrotor with respect to a ground target. The relative position of the quadrotor will be measured in the vehicle-1 frame, i.e., the vehicle frame after it has been rotated by the heading vector ψ. The vehicle-1 frame is convenient since x, y, and z positions are still measured relative to a flat earth, but they are vehicle centered quantities as opposed to inertial quantities. Let p_x, p_y, and p_z denote the relative position vector between the target and the vehicle resolved in the $v1$ frame. Therefore Eq (16) becomes

$$\begin{pmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{p}_z \end{pmatrix} = \begin{pmatrix} c\theta & s\phi s\theta & c\phi s\theta \\ 0 & c\phi & -s\phi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{pmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix}. \tag{20}$$

4.2 Model for control design

Assuming that ϕ and θ are small, Equation (18) can be simplified as

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} p \\ q \\ r \end{pmatrix}. \tag{21}$$

Similarly, Equation (19) is simplified by assuming that the Coriolis terms qr, pr, and pq, are small to obtain

$$\begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} = \begin{pmatrix} \frac{1}{J_x}\tau_\phi \\ \frac{1}{J_y}\tau_\theta \\ \frac{1}{J_z}\tau_\psi \end{pmatrix}. \tag{22}$$

Combining Eq. (21) and (22) we get

$$\begin{pmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{pmatrix} = \begin{pmatrix} \frac{1}{J_x}\tau_\phi \\ \frac{1}{J_y}\tau_\theta \\ \frac{1}{J_z}\tau_\psi \end{pmatrix}. \tag{23}$$

Differentiating Eq. (16) and neglecting $\dot{R}_b^v$ gives

$$\begin{pmatrix} \ddot{p}_n \\ \ddot{p}_e \\ \ddot{p}_d \end{pmatrix} = \begin{pmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{pmatrix} \begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix}. \tag{24}$$

Neglecting the Coriolis terms and plugging Eq. (17) into Eq. (24) and simplifying gives

$$\begin{pmatrix} \ddot{p}_n \\ \ddot{p}_e \\ \ddot{p}_d \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ g \end{pmatrix} + \begin{pmatrix} -c\phi s\theta c\psi - s\phi s\psi \\ -c\phi s\theta s\psi + s\phi c\psi \\ -c\phi c\theta \end{pmatrix} \frac{F}{m}. \tag{25}$$

Therefore, the simplified inertial model is given by

$$\ddot{p}_n = (-\cos\phi \sin\theta \cos\psi - \sin\phi \sin\psi) \frac{F}{m} \tag{26}$$

$$\ddot{p}_e = (-\cos\phi \sin\theta \sin\psi + \sin\phi \cos\psi) \frac{F}{m} \tag{27}$$

$$\ddot{p}_d = g - (\cos\phi \cos\theta) \frac{F}{m} \tag{28}$$

$$\ddot{\phi} = \frac{1}{J_x} \tau_\phi \tag{29}$$

$$\ddot{\theta} = \frac{1}{J_y} \tau_\theta \tag{30}$$

$$\ddot{\psi} = \frac{1}{J_z} \tau_\psi. \tag{31}$$

$$\tag{32}$$

The dynamics given in Equations (26)–(31) are expressed in the inertial frame. This is necessary for the simulator. However, we will be controlling position, altitude, and heading using camera frame measurements of a target position. In this setting heading is irrelevant. Therefore, instead of expressing the translational dynamics in the inertial frame, we will express them in the vehicle-1 frame $\mathcal{F}^{v1}$, which is equivalent to the inertial frame after rotating by the heading angle.

Differentiating Eq. (20) and neglecting $\dot{R}_b^{v1}$ gives

$$\begin{pmatrix} \ddot{p}_x \\ \ddot{p}_y \\ \ddot{p}_z \end{pmatrix} = \begin{pmatrix} c\theta & s\phi s\theta & c\phi s\theta \\ 0 & c\phi & -s\phi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{pmatrix} \begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix}. \tag{33}$$

Neglecting the Coriolis terms and plugging Eq. (17) into Eq. (33) and simplifying gives

$$\begin{pmatrix} \ddot{p}_x \\ \ddot{p}_y \\ \ddot{p}_z \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ g \end{pmatrix} + \begin{pmatrix} -c\phi s\theta \\ s\phi \\ -c\phi c\theta \end{pmatrix} \frac{F}{m}. \tag{34}$$

Therefore, the simplified model in the vehicle-1 frame is given by

$$\ddot{p}_x = -\cos\phi \sin\theta \frac{F}{m} \tag{35}$$

$$\ddot{p}_y = \sin\phi \frac{F}{m} \tag{36}$$

$$\ddot{p}_z = g - \cos\phi\cos\theta \frac{F}{m} \tag{37}$$

$$\ddot{\phi} = \frac{1}{J_x}\tau_\phi \tag{38}$$

$$\ddot{\theta} = \frac{1}{J_y}\tau_\theta \tag{39}$$

$$\ddot{\psi} = \frac{1}{J_z}\tau_\psi. \tag{40}$$

$$\tag{41}$$

5 Sensors

5.1 Rate Gyros

A MEMS rate gyro contains a small vibrating lever. When the lever undergoes an angular rotation, Coriolis effects change the frequency of the vibration, thus detecting the rotation. A brief description of the physics of rate gyros can be found at RateSensorAppNote.pdf.

The output of the rate gyro is given by

$$y_{gyro} = k_{gyro}\Omega + \beta_{gyro} + \eta_{gyro},$$

where y_{gyro} is in Volts, k_{gyro} is a gain, Ω is the angular rate in radians per second, β_{gyro} is a bias term, and η_{gyro} is zero mean white noise. The gain k_{gyro} should be given on the spec sheet of the sensor. However, due to variations in manufacturing it is imprecisely known. The bias term β_{gyro} is strongly dependent on temperature and should be calibrated prior to each flight.

If three rate gyros are aligned along the x, y, and z axes of the quadrotor, then the rate gyros measure the angular body rates p, q, and r as follows:

$$\begin{aligned} y_{gyro,x} &= k_{gyro,x} p + \beta_{gyro,x} + \eta_{gyro,x} \\ y_{gyro,y} &= k_{gyro,y} q + \beta_{gyro,y} + \eta_{gyro,y} \\ y_{gyro,z} &= k_{gyro,z} r + \beta_{gyro,z} + \eta_{gyro,z}. \end{aligned}$$

MEMS gyros are analog devices that are sampled by the on-board processer. We will assume that the sample rate is given by T_s. The Kestrel autopilot samples the gyros at approximately 120 Hz.

5.2 Accelerometers

A MEMS accelerometer contains a small plate attached to torsion levers. The plate rotates under acceleration and changes the capacitance between the plate and the surrounding walls [3].

The output of the accelerometers is given by

$$y_{acc} = k_{acc} A + \beta_{acc} + \eta_{acc},$$

where y_{acc} is in Volts, k_{acc} is a gain, A is the acceleration in meters per second, β_{acc} is a bias term, and η_{acc} is zero mean white noise. The gain k_{acc} should be given on the spec sheet of the sensor. However, due to variations in manufacturing it is imprecisely known. The bias term β_{acc} is strongly dependent on temperature and should be calibrated prior to each flight.

Accelerometers measure the specific force in the body frame of the vehicle. A physically intuitive explanation is given in [1, p. 13-15]. Additional explanation is given in [4, p. 27]. Mathematically we have

$$\begin{aligned} \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} &= \frac{1}{m} \left(\mathbf{F} - \mathbf{F}_{\text{gravity}} \right) \\ &= \dot{\mathbf{v}} + \boldsymbol{\omega}^b_{b/i} \times \mathbf{v} - \frac{1}{m} \mathbf{F}_{\text{gravity}}. \end{aligned}$$

In component form we have

$$\begin{aligned} a_x &= \dot{u} + qw - rv + g \sin\theta \\ a_y &= \dot{v} + ru - pw - g \cos\theta \sin\phi \\ a_z &= \dot{w} + pv - qu - g \cos\theta \cos\phi. \end{aligned}$$

Using Eq (17) we get

$$\begin{aligned} a_x &= 0 \\ a_y &= 0 \\ a_z &= -F/m, \end{aligned}$$

where F is the total thrust produced by the four motors. It is important to note that for the quadrotor, the output of the accelerometers is independent of angle. This is in contrast to the unaccelerated flight for fixed wing vehicles where the accelerometers are used to measure the gravity vector and thereby extract roll and pitch angles.

MEMS accelerometers are analog devices that are sampled by the on-board processer. We will assume that the sample rate is given by T_s. The Kestrel autopilot samples the accelerometers at approximately 120 Hz.

5.3 Camera

The control objective is to hold the position of the quadrotor over a ground based target that is detected using the vision sensor. In this section we will briefly describe how to estimate p_x and p_y in the vehicle 1-frame.

We will assume that the camera is mounted so that the optical axis of the camera is aligned with the body frame z-axis and so that the x-axis of the camera points out the right of the quadrotor and the y-axis of the camera points to the back of the quadrotor.

The camera model is shown in Figure 14. The position of the target in the vehicle-1 frame is (p_x, p_y, p_z). The pixel location of the target in the image is (ϵ_x, ϵ_y).

The geometry for p_y is shown in Figure 15. From the geometry shown in Figure 15, we can see that

$$p_y = p_z \tan\left(\phi - \epsilon_x \frac{\eta}{M_y}\right), \tag{42}$$

where η is the camera field-of-view, and M_y is the number of pixels along the camera y-axis. In Figure 15, both p_y and ϵ_x are negative. Positive values are toward the right rotor. A similar equation can be derived for p_x as

$$p_x = -p_z \tan\left(\theta - \epsilon_y \frac{\eta}{M_y}\right). \tag{43}$$

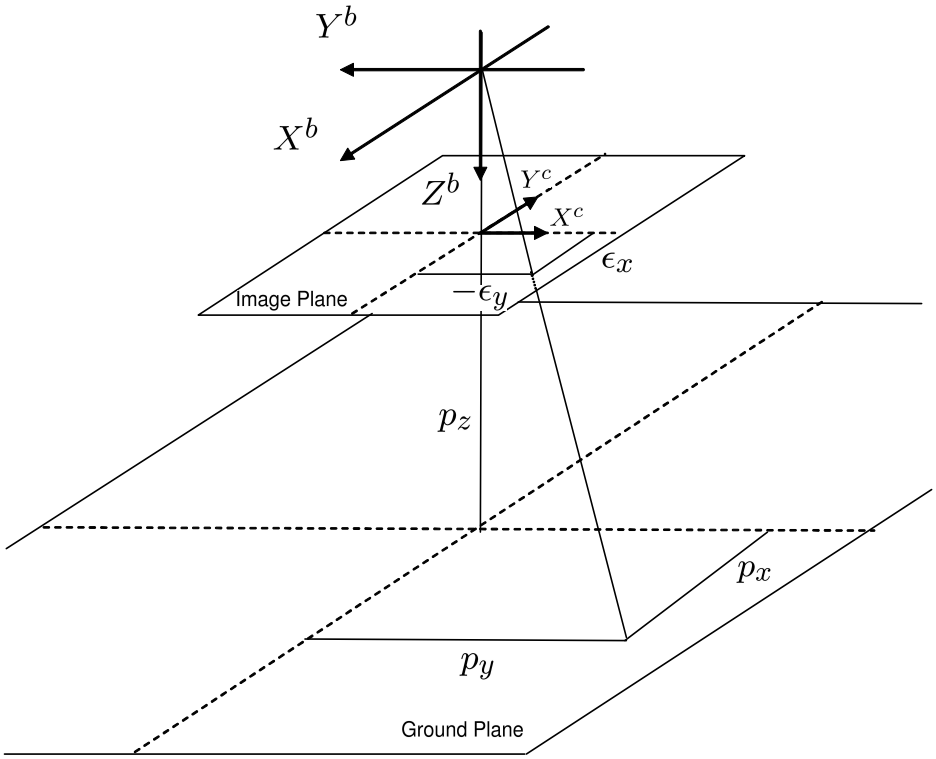

Figure 14: Camera model for the quadrotor.

6 State Estimation

The objective of this section is to describe techniques for estimating the state of the quadrotor from sensor measurements. We need to estimate the following states: p_x, p_y, p_z, u, v, w, ϕ, θ, ψ, p, q, r.

The angular rates p, q, and r can be obtained by low pass filtering the rate gyros. The remain states require a Kalman filter. Both are discussed below.

6.1 Low Pass Filters

The Laplace transforms representation of a simple low-pass filter is given by

$$Y(s) = \frac{a}{s+a}U(s),$$

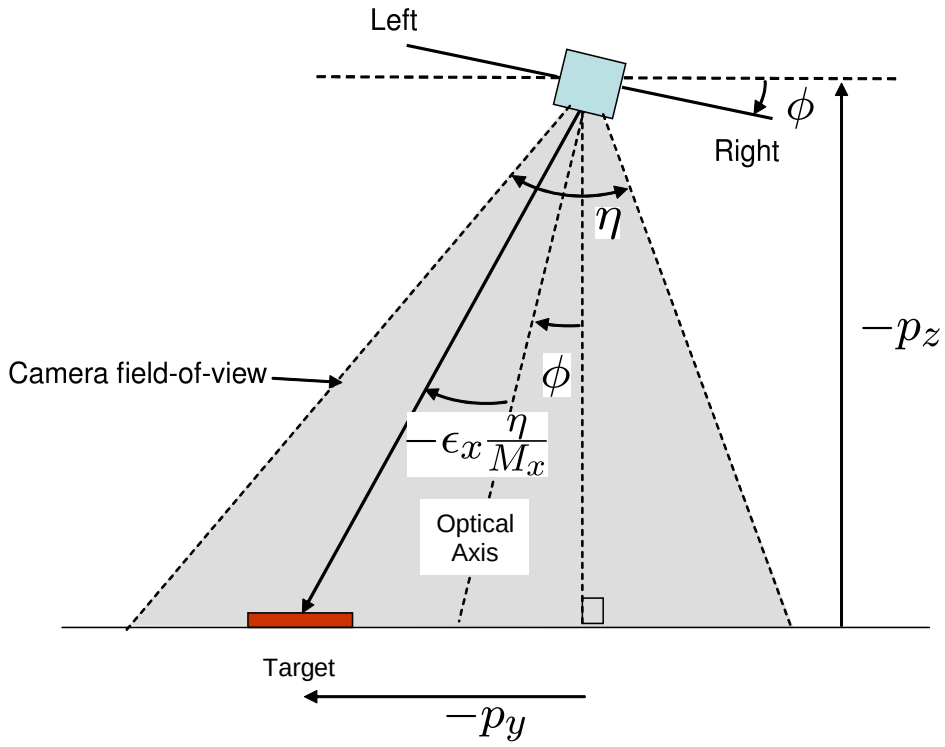

Figure 15: The geometry introduced by the vision system. The height above ground is given by $-p_z$, the lateral position error is p_y, the roll angle is ϕ, the field-of-view of the camera is η, the lateral pixel location of the target in the image is ϵ_x, and the total number of pixels along the lateral axis of the camera is M_x.

were $u(t)$ is the input of the filter and $y(t)$ is the output. Inverse Laplace transforming we get

$$\dot{y} = -ay + au. \tag{44}$$

Using a zeroth order approximation of the derivative we get

$$\frac{y(t+T) - y(t)}{T} = -ay(t) + au(t),$$

where T is the sample rate. Solving for $y(t+T)$ we get

$$y(t+T) = (1 - aT)y(t) + aTu(t).$$

For the zeroth order approximation to be valid we need $aT \ll 1$. If we let $\alpha = aT$ then we get the simple form

$$y(t+T) = (1 - \alpha)y(t) + \alpha u(t).$$

Note that this equation has a nice physical interpretation: the new value of y (filtered value) is a weighted average of the old value of y and u (unfiltered value). If u is noisy, then $\alpha \in [0, 1]$ should be set to a small value. However, if u is relatively noise free, then α should be close to unity.

In the derivation of the discrete-time implementation of the low-pass filter, it is possible to be more precise. In particular, returning to Equation (44), from linear systems theory, it is well known that the solution is given by

$$y(t+T) = e^{-aT}y(t) + a\int_0^T e^{-a(T-\tau)}u(\tau)\,d\tau.$$

Assuming that $u(t)$ is constant between sample periods results in the expression

$$\begin{aligned} y(t+T) &= e^{-aT}y(t) + a\int_0^T e^{-a(T-\tau)}\,d\tau u(t)y(t+T) \\ &= e^{-aT}y(t) + (1-e^{-aT})u(t). \end{aligned} \tag{45}$$

Note that since $e^x = 1 + x + \frac{x^2}{2!} + \ldots$ we have that $e^{-aT} \approx 1 - aT$ and $1 - e^{-aT} \approx aT$.

Therefore, if the sample rate is not fixed (common for micro controllers), and it is desired to have a fixed cut-off frequency, then Equation (45) is the preferable way to implement a low-pass filter in digital hardware.

We will use the notation $LPF(\cdot)$ to represent the low-pass filter operator. Therefore $\hat{x} = LPF(x)$ is the low-pass filtered version of x.

6.2 Angular Rates p, q, and r.

The angular rates p, q, and r can be estimated by low-pass filtering the rate gyro signals:

$$\hat{p} = LPF(y_{\text{gyro,x}}) \tag{46}$$
$$\hat{q} = LPF(y_{\text{gyro,y}}) \tag{47}$$
$$\hat{r} = LPF(y_{\text{gyro,z}}). \tag{48}$$

6.3 Dynamic Observer Theory

The objective of this section is to briefly review observer theory.

Suppose that we have a linear time-invariant system modeled by the equations

$$\dot{x} = Ax + Bu$$
$$y = Cx.$$

A continuous-time observer for this system is given by the equation

$$\dot{\hat{x}} = \underbrace{A\hat{x} + Bu}_{\text{copy of the model}} + \underbrace{L\,(y - C\hat{x})}_{\text{correction due to sensor reading}}, \tag{49}$$

where $\hat{x}$ is the estimated value of x. Letting $\tilde{x} = x - \hat{x}$ we observer that

$$\dot{\tilde{x}} = (A - LC)\tilde{x}$$

which implies that the observation error decays exponentially to zero if L is chosen such that the eigenvalues of $A - LC$ are in the open left half of the complex plane.

In practice, the sensors are usually sampled and processed in digital hardware at some sample rate T_s. How do we modify the observer equation shown in Equation (49) to account for sampled sensor readings?

The typical approach is to propagate the system model between samples using the equation

$$\dot{\hat{x}} = A\hat{x} + Bu \tag{50}$$

and then to update the estimate when a measurement is received using the equation

$$\hat{x}^+ = \hat{x}^- + L(y(t_k) - C\hat{x}^-),$$

where t_k is the instant in time that the measurement is received and $\hat{x}^-$ is the state estimate produced by Equation (50) at time t_k. Equation (50) is then re-instantiated with initial conditions given by $\hat{x}^+$. The continuous-discrete observer is summarized in Table 6.3.

The observation process is shown graphically in Figure 16. Note that it is not necessary to have a fixed sample rate. The continuous-discrete observer can be implemented using Algorithm 1.

Note that we did not use the fact that the process was linear. Suppose instead that we have a nonlinear system of the form

$$\dot{x} = f(x, u) \tag{51}$$
$$y = c(x) \tag{52}$$

, then the continuous discrete observer is given in table 6.3.

The real question is how to pick the observer gain L.

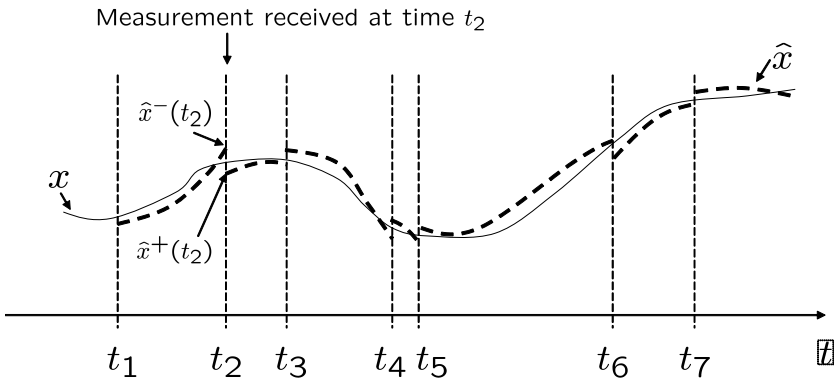

Figure 16: sdf

Algorithm 1 Continuous-Discrete Observer

1: Initialize: $\hat{x} = 0$.
2: Pick an output sample rate T_{out} which is much less than the sample rates of the sensors.
3: At each sample time T_{out}:
4: **for** $i = 1$ to N **do** {Prediction: Propagate the state equation.}
5: $\quad \hat{x} = \hat{x} + \left(\frac{T_{out}}{N}\right)(A\hat{x} + Bu)$
6: **end for**
7: **if** A measurement has been received from sensor i **then** {Correction: Measurement Update}
8: $\quad \hat{x} = \hat{x} + L_i\left(y_i - C_i\hat{x}\right)$
9: **end if**

System model:

$\dot{x} = Ax + Bu$

$y(t_k) = Cx(t_k)$

Initial Condition $x(0)$.

Assumptions:

Knowledge of A, B, C, $u(t)$.

No measurement noise.

Prediction: In between measurements $(t \in [t_{k-1}, t_k))$**:**

Propagate $\dot{\hat{x}} = A\hat{x} + Bu$.

Initial condition is $\hat{x}^+(t_{k-1})$.

Label the estimate at time t_k as $\hat{x}^-(t_k)$.

Correction: At sensor measurement $(t = t_k)$**:**

$\hat{x}^+(t_k) = \hat{x}^-(t_k) + L\left(y(t_k) - C\hat{x}^-(t_k)\right).$

Table 1: Continuous-Discrete Linear Observer.

6.4 Essentials from Probability Theory

Let $X = (x_1, \ldots, x_n)^T$ be a random vector whose elements are random variables. The mean, or expected value of X is denoted by

$$\boldsymbol{\mu} = \begin{pmatrix} \mu_1 \\ \vdots \\ \mu_n \end{pmatrix} = \begin{pmatrix} E\{x_1\} \\ \vdots \\ E\{x_n\} \end{pmatrix} = E\{X\},$$

where

$$E\{x_i\} = \int \xi f_i(\xi)\, d\xi,$$

and $f(\cdot)$ is the probability density function for x_i. Given any pair of components x_i and x_j of X, we denote their covariance as

$$\textit{cov}(x_i, x_j) = \Sigma_{ij} = E\{(x_i - \mu_i)(x_j - \mu_j)\}.$$

The covariance of any component with itself is the variance, i.e.,

$$\textit{var}(x_i) = cov(x_i, x_i) = \Sigma_{ii} = E\{(x_i - \mu_i)(\xi - \mu_i)\}.$$

The standard deviation of x_i is the square root of the variance:

$$\textit{stdev}(x_i) = \sigma_i = \sqrt{\Sigma_{ii}}.$$

System model:

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

$y(t_k) = c(x(t_k))$

Initial Condition $x(0)$.

Assumptions:

Knowledge of f, c, $u(t)$.

No measurement noise.

Prediction: In between measurements $(t \in [t_{k-1}, t_k))$**:**

Propagate $\dot{\hat{x}} = f(\hat{x}, u)$.

Initial condition is $\hat{x}^+(t_{k-1})$.

Label the estimate at time t_k as $\hat{x}^-(t_k)$.

Correction: At sensor measurement $(t = t_k)$**:**

$\hat{x}^+(t_k) = \hat{x}^-(t_k) + L\left(y(t_k) - c(\hat{x}^-(t_k))\right).$

Table 2: Continuous-Discrete Nonlinear Observer.

The covariances associated with a random vector X can be grouped into a matrix known as the covariance matrix:

$$\Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} & \cdots & \Sigma_{1n} \\ \Sigma_{21} & \Sigma_{22} & \cdots & \Sigma_{2n} \\ \vdots & & \ddots & \vdots \\ \Sigma_{n1} & \Sigma_{n2} & \cdots & \Sigma_{nn} \end{pmatrix} = E\{(X - \boldsymbol{\mu})(X - \boldsymbol{\mu})^T\} = E\{XX^T\} - \boldsymbol{\mu}\boldsymbol{\mu}^T.$$

Note that $\Sigma = \Sigma^T$ so that Σ is both symmetric and positive semi-definite, which implies that its eigenvalues are real and nonnegative.

The probability density function for a Gaussian random variable is given by

$$f_x(x) = \frac{1}{\sqrt{2\pi}\sigma_x} e^{-\frac{(x-\mu_x)^2}{\sigma_x^2}},$$

where μ_x is the mean of x and σ_x is the standard deviation. The vector equivalent is given by

$$f_X(X) = \frac{1}{\sqrt{2\pi \det \Sigma}} \exp\left[-\frac{1}{2}(X - \boldsymbol{\mu})^T \Sigma^{-1} (X - \boldsymbol{\mu})\right],$$

in which case we write

$$X \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma),$$

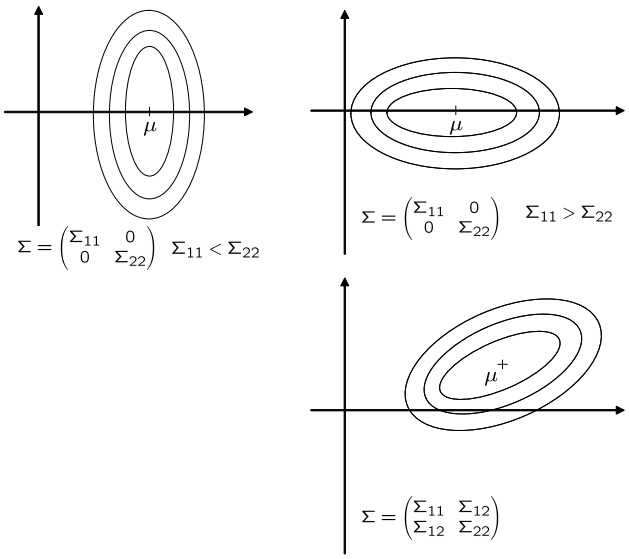

Figure 17: Level curves for the pdf of a 2D Gaussian random variable.

and say that X is normally distributed with mean $\boldsymbol{\mu}$ and covariance Σ.

Figure 17 shows the level curves for a 2D Gaussian random variable with different covariance matrices.

6.5 Derivation of the Kalman Filter

In this section we assume the following state model:

$$\begin{aligned} \dot{x} &= Ax + Bu + G\xi \\ y_k &= Cx_k + \eta_k, \end{aligned}$$

where $y_k = y(t_k)$ is the k^{th} sample of y, $x_k = x(t_k)$ is the k^{th} sample of x, η_k is the measurement noise at time t_k, ξ is a zero-mean Gaussian random process with covariance Q, and η_k is a zero-mean Gaussian random variable with covariance R. Note that the sample rate does not need to be be fixed.

The observer will therefore have the following form:

Prediction: In between measurements: $(t \in [t_{k-1}, t_k])$
Propagate $\dot{\hat{x}} = A\hat{x} + Bu$.
Initial condition is $\hat{x}^+(t_{k-1})$.
Label the estimate at time t_k as $\hat{x}^-(t_k)$.
Correction: At sensor measurement $(t = t_k)$**:**
$\hat{x}^+(t_k) = \hat{x}^-(t_k) + L\left(y(t_k) - C\hat{x}^-(t_k)\right).$
Our objective is to pick L to minimize tr$(P(t))$.

6.5.1 Between Measurements.

Differentiating $\tilde{x}$ we get

$$\begin{aligned}\dot{\tilde{x}} &= \dot{x} - \dot{\hat{x}} \\ &= Ax + Bu + G\xi - A\hat{x} - Bu \\ &= A\tilde{x} + G\xi.\end{aligned}$$

Therefore we have that

$$\tilde{x}(t) = e^{At}\tilde{x}_0 + \int_0^t e^{A(t-\tau)}G\xi(\tau)\,d\tau.$$

We can therefore compute the evolution for P as

$$\begin{aligned}\dot{P} &= \frac{d}{dt}E\{\tilde{x}\tilde{x}^T\} \\ &= E\{\dot{\tilde{x}}\tilde{x}^T + \tilde{x}\dot{\tilde{x}}^T\} \\ &= E\left\{A\tilde{x}\tilde{x}^T + G\xi\tilde{x}^T + \tilde{x}\tilde{x}^TA^T + \tilde{x}\xi^TG^T\right\} \\ &= AP + PA^T + GE\{\xi\tilde{x}^T\}^T + E\{\tilde{x}\xi^T\}G^T.\end{aligned}$$

As in the previous section we get

$$\begin{aligned}E\{\xi\tilde{x}^T\} &= E\left\{\xi(t)\tilde{x}_0e^{A^Tt} + \int_0^t \xi(t)\xi^T(\tau)G^Te^{A^T(t-\tau)}\,d\tau\right\} \\ &= \frac{1}{2}QG^T,\end{aligned}$$

which implies that

$$\dot{P} = AP + PA^T + GQG^T.$$

6.5.2 At Measurements.

At a measurement we have that

$$\begin{aligned}\tilde{x}^+ &= x - \hat{x}^+ \\ &= x - \hat{x}^- - L\left(Cx + \eta - C\hat{x}^-\right) \\ &= \tilde{x}^- - LC\tilde{x}^- - L\eta.\end{aligned}$$

Therefore

$$\begin{aligned}P^+ &= E\{\tilde{x}^+\tilde{x}^{+T}\} \\ &= E\left\{\left(\tilde{x}^- - LC\tilde{x}^- - L\eta\right)\left(\tilde{x}^- - LC\tilde{x}^- - L\eta\right)^T\right\} \\ &= E\left\{\tilde{x}^-\tilde{x}^{-T} - \tilde{x}^-\tilde{x}^{-T}C^TL^T - \tilde{x}^-\eta^TL^T\right. \\ &\quad - LC\tilde{x}^-\tilde{x}^{-T} + LC\tilde{x}^-\tilde{x}^{-T}C^TL^T + LC\tilde{x}^-\eta^TL^T \\ &= \quad -L\eta\tilde{x}^{-T} + L\eta\tilde{x}^{-T}C^TL^T + L\eta\eta^TL^T\Big\} \\ &= P^- - P^-C^TL^T - LCP^- + LCP^-C^TL^T + LRL^T. \qquad (53)\end{aligned}$$

Our objective is to pick L to minimize $\text{tr}(P^+)$. A necessary condition is

$$\begin{aligned}\frac{\partial}{\partial L}\text{tr}(P^+) &= -P^-C^T - P^-C^T + 2LCP^-C^T + 2LR = 0 \\ &\implies 2L(R + CP^-C^T) = 2P^-C^T \\ &\implies L = P^-C^T(R + CP^-C^T)^{-1}.\end{aligned}$$

Plugging back into Equation (53) give

$$\begin{aligned}P^+ &= P^- + P^-C^T(R + CP^-C^T)^{-1}CP^- - P^-C^T(R + CP^-C^T)^{-1}CP^- \\ &\quad + P^-C^T(R + CP^-C^T)^{-1}(CP^-C^T + R)(R + CP^-C^T)^{-1}CP^- \\ &= P^- - P^-C^T(R + CP^-C^T)^{-1}CP^- \\ &= (I - P^-C^T(R + CP^-C^T)^{-1}C)P^- \\ &= (I - LC)P^-.\end{aligned}$$

For linear systems, the continuous-discrete Kalman filter is summarized in Table 6.5.2

If the system is nonlinear, then the Kalman filter can still be applied but we need to linearize the nonlinear equations in order to compute the error covariance matrix P and the Kalman gain L. The extended Kalman filter (EKF) is given in Table 6.5.2, and an algorithm to implement the EKF is given in Algorithm 2.

System model:

$\dot{x} = Ax + Bu + \xi$

$y_i(t_k) = C_i x(t_k) + \eta_k$

Initial Condition $x(0)$.

Assumptions:

Knowledge of A, B, C_i, $u(t)$.

Process noise satisfies $\xi \sim \mathcal{N}(0, Q)$.

Measurement noise satisfies $\eta_k \sim \mathcal{N}(0, R)$.

Prediction: In between measurements $(t \in [t_{k-1}, t_k))$**:**

Propagate $\dot{\hat{x}} = A\hat{x} + Bu$.

Propagate $\dot{P} = AP + PA^T + Q$.

Correction: At the i^{th} **sensor measurement** $(t = t_k)$**:**

$L_i = P^- C_i^T (R_i + C_i P C_i^T)^{-1}$,

$P^+ = (I - L_i C_i) P^-$,

$\hat{x}^+(t_k) = \hat{x}^-(t_k) + L\left(y(t_k) - C\hat{x}^-(t_k)\right)$.

Table 3: Continuous-Discrete Kalman Filter.

6.6 Application to the quadrotor

In this section we will discuss the application of Algorithm 2 to the quadrotor.

We would like to estimate the state

$$\hat{x} = \begin{pmatrix} \hat{p}_x \\ \hat{p}_y \\ \hat{p}_z \\ \dot{\hat{p}}_x \\ \dot{\hat{p}}_y \\ \dot{\hat{p}}_z \\ \hat{\phi} \\ \hat{\theta} \\ \hat{\psi} \end{pmatrix},$$

where the rate gyros and accelerometers will be used to drive the prediction step, and an ultrasonic altimeter and camera will be used in the correction step.

System model:

$\dot{x} = f(x,u) + \xi$

$y_i(t_k) = c_i(x(t_k)) + \eta_k$

Initial Condition $x(0)$.

Assumptions:

Knowledge of f, c_i, $u(t)$.

Process noise satisfies $\xi \sim \mathcal{N}(0, Q)$.

Measurement noise satisfies $\eta_k \sim \mathcal{N}(0, R)$.

Prediction: In between measurements $(t \in [t_{k-1}, t_k))$**:**

Propagate $\dot{\hat{x}} = f(\hat{x}, u)$,

Compute $A = \frac{\partial f}{\partial x}\big|_{x=\hat{x}(t)}$,

Propagate $\dot{P} = AP + PA^T + Q$.

Correction: At the i^{th} **sensor measurement** $(t = t_k)$**:**

$C_i = \frac{\partial c_i}{\partial x}\big|_{x=\hat{x}^-}$

$L_i = P^- C_i^T (R_i + C_i P C_i^T)^{-1}$,

$P^+ = (I - L_i C_i) P^-$,

$\hat{x}^+(t_k) = \hat{x}^-(t_k) + L\left(y(t_k) - c_i(\hat{x}^-(t_k))\right)$.

Table 4: Continuous-Discrete Extended Kalman Filter.

The propagation model is obtained from Equations (35)–(37), and (18) as

$$f(x,u) = \begin{pmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{p}_z \\ \cos\phi \sin\theta a_z \\ -sin\phi a_z \\ g + \cos\phi\cos\theta a_z \\ p + q\sin\phi\tan\theta + r\cos\phi\tan\theta \\ q\cos\phi - r\sin\phi \\ q\frac{\sin\phi}{\cos\theta} + r\frac{\cos\phi}{\cos\theta} \end{pmatrix},$$

where we have used the fact that the z-axis of the accelerometer measures $a_z =$

Algorithm 2 Continuous-Discrete Extended Kalman Filter

Initialize: $\hat{x} = 0$.
Pick an output sample rate T_{out} which is much less than the sample rates of the sensors.
At each sample time T_{out}:
for $i = 1$ to N **do** {Prediction: Propagate the equations.}
  $\hat{x} = \hat{x} + \left(\frac{T_{out}}{N}\right)(f(\hat{x}, u))$
  $A = \frac{\partial f}{\partial x}$
  $P = P + \left(\frac{T_{out}}{N}\right)\left(AP + PA^T + GQG^T\right)$
end for
if A measurement has been received from sensor i **then** {Correction: Measurement Update}
  $C_i = \frac{\partial c_i}{\partial x}$
  $L_i = PC_i^T(R_i + C_iPC_i^T)^{-1}$
  $P = (I - L_iC_i)P$
  $\hat{x} = \hat{x} + L_i\left(y_i - c_i(\hat{x})\right)$.
end if

$-F/m$. Differentiating we obtain

$$A = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -s_\phi s_\theta a_z & c_\phi c_\theta a_z & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -c_\phi a_z & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -s_\phi c_\theta a_z & -c_\phi s_\theta a_z & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & qc\phi t\theta - rs\phi t\theta & (qs\phi + rc\phi)/c^2\theta & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -qs\phi - rc\phi & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{qc\phi - rs\phi}{c\theta} & -(qs\phi + rc\phi)\frac{t\theta}{c\theta} & 0 \end{pmatrix}$$

Note that it may be adequate (not sure) to use a small angle approximation in

the model resulting in

$$f(x,u) = \begin{pmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{p}_z \\ \theta a_z \\ -\phi a_z \\ g + a_z \\ p \\ q \\ r \end{pmatrix},$$

and

$$A = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & a_z & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -a_z & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

If this form works, then the update equation for P can be coded by hand, significantly reducing the computational burden. Note also that $f(x,u)$ does not take into account the motion of the target. A feedforward term can be added to f to account for the target motion as

$$f(x,u) = \begin{pmatrix} \dot{p}_x - \dot{m}_x \\ \dot{p}_y - \dot{m}_y \\ \dot{p}_z \\ \theta a_z - \ddot{m}_x \\ -\phi a_z - \ddot{m}_y \\ g + a_z \\ p \\ q \\ r \end{pmatrix},$$

where $(\dot{m}_x, \dot{m}_y)$ is the velocity of the target in the vehicle-1 frame, and $(\ddot{m}_x, \ddot{m}_y)$ is the acceleration of the target in the vehicle-1 frame.

Let the relative position between the quadrotor and the target be denoted by $\mathbf{p}^{v1} = (p_x, p_y, p_z)^T$. We can tranform to the camera frame as

$$\mathbf{p}^c = R_g^c R_b^g R_{v1}^b \mathbf{p}^{v1},$$

where

$$R_g^c = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

is the rotation matrix from gimbal coordinates to camera coordinates,

$$R_b^g = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix}$$

transforms body coordinates to gimbal coordinates, and

$$R_{v1}^b = \begin{pmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{pmatrix}$$

transforms vehicle-1 coordinates to body coordinates. Using the small angle approximation for ϕ and θ we get

$$\mathbf{p}^c = \begin{pmatrix} \phi\theta p_x + p_y + \phi p_z \\ -p_x + \theta p_z \\ \theta p_x - \phi p_y + p_z \end{pmatrix}.$$

The model for the pixel coordinates are therefore computed as

$$\epsilon_x = c_{\epsilon_x}(x) \triangleq f \frac{\phi\theta p_x + p_y + \phi p_z}{\theta p_x - \phi p_y + p_z}$$
$$\epsilon_2 = c_{\epsilon_y}(x) \triangleq f \frac{-p_x + \theta p_z}{\theta p_x - \phi p_y + p_z}.$$

We therefore have the following sensors available to correct the state estimate:

$$\text{sensor 1: altimeter} \quad y_1 = c_{alt}(x) \triangleq -p_z + \eta_1[k] \tag{54}$$
$$\text{sensor 2: vision x-pixel} \quad y_2 = c_{\epsilon_x}(x) + \eta_2[k]. \tag{55}$$
$$\text{sensor 3: vision y-pixel} \quad y_3 = c_{\epsilon_y}(x) + \eta_3[k] \tag{56}$$
$$\text{sensor 4: vision heading} \quad y_4 = c_{\epsilon_\psi}(x) \triangleq \pi/2 + \psi + \eta_4[k]. \tag{57}$$

The linearization of the first and fourth output functions are given by

$$C_1 = \begin{pmatrix} 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{58}$$

$$C_4 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \tag{59}$$

The linearization of the expression for the pixel coordinates is messy and can easily be computed numerically using the approximation

$$\frac{\partial f(x_1, \cdots, x_i, \ddots, x_n)}{\partial x_i} \approx \frac{f(x_1, \cdots, x_i + \Delta, \cdots, x_n) - f(x_1, \cdots, x_i, \cdots, x_n)}{\Delta},$$

where Δ is a small constant.

7 Control Design

The control design will be derived directly from Equations (35)–(40). Equations (38)–(40) are already linear. To simplify Equations (35)–(37) define

$$u_x \stackrel{\triangle}{=} -cos\phi \sin\theta \frac{F}{m} \tag{60}$$

$$u_y \stackrel{\triangle}{=} \sin\phi \frac{F}{m} \tag{61}$$

$$u_z \stackrel{\triangle}{=} g - \cos\phi \cos\theta \frac{F}{m}, \tag{62}$$

to obtain

$$\ddot{p}_x = u_x \tag{63}$$

$$\ddot{p}_y = u_y \tag{64}$$

$$\ddot{p}_z = u_z. \tag{65}$$

The control design proceeds by developing PID control strategies for u_x, u_y, and u_z. After u_x, u_y, and u_z have been computed, we can compute the desired force F, and the commanded roll angle ϕ^c and the commanded pitch angle θ^c from Equations (60)–(62) as follows. From Equation (62) solve for F/m as

$$\frac{F}{m} = \frac{g - u_z}{\cos\phi \cos\theta}. \tag{66}$$

Substituting (66) into (61) gives

$$u_y = \frac{g - u_z}{\cos\theta} \tan\phi.$$

Solving for ϕ and letting this be the commanded roll angle gives

$$\phi^c = \tan^{-1}\left(\frac{u_y \cos\theta}{g - u_z}\right). \tag{67}$$

Similarly, we can solve for the commanded pitch angle as

$$\theta^c = \tan^{-1}\left(\frac{u_x}{u_z - g}\right). \tag{68}$$

7.1 Vision Based Altitude Hold

For the altitude hold we need to develop an expression for u_z to drive p_z to a desired altitude based on the size of the object in the image. We will assume that the camera returns the size of the object in the image plane in pixels, which is denoted by ϵ_s. Figure 18 shows the geometry of the quadrotor hovering over a

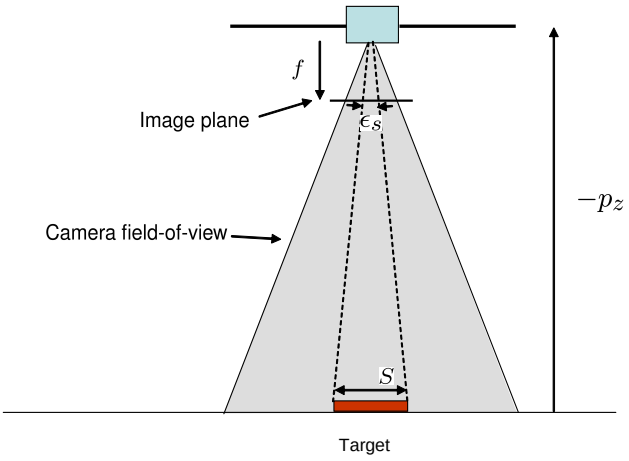

Figure 18: The size of the target is S in meters, and the size of the target in the image plane is denoted by ϵ_s in pixels. The focal length is f, and the height above the target is $-p_z$.

target of size S. From similar triangles we have that

$$\frac{-p_z}{S} = \frac{f}{\epsilon_s}.$$

Solving for p_z an differentiating we obtain

$$\dot{p}_z = \frac{fS\dot{\epsilon}_s}{\epsilon_s^2}. \tag{69}$$

Differentiating again gives

$$\ddot{p}_z = \frac{fS\ddot{\epsilon}_s}{\epsilon_s^2} - 2fS\frac{\dot{\epsilon}_s^2}{\epsilon_s^3}.$$

Substituting $u_z = \ddot{p}_z$ and solving for $\ddot{\epsilon}_s$ gives

$$\ddot{\epsilon}_s = \left(\frac{\epsilon_s^2}{fS}\right) u_z + 2\frac{\dot{\epsilon}_s^2}{\epsilon_s}.$$

Defining

$$u_s \triangleq \left(\frac{\epsilon_s^2}{fS}\right) u_z + 2\frac{\dot{\epsilon}_s^2}{\epsilon_s}, \tag{70}$$

we get

$$\ddot{\epsilon}_s = u_s.$$

We can now derive a PID controller to drive $\epsilon_s \rightarrow \epsilon_s^d$ as

$$u_s = k_{p_s}(\epsilon_s^d - \epsilon_s) - k_{d_s}\dot{\epsilon}_s + k_{i_s}\int_0^t (\epsilon_s^d - \epsilon_s)\, d\tau.$$

Solving (70) for u_z we get

$$u_z = \frac{fS}{\epsilon_s^2}k_{p_s}(\epsilon_s^d - \epsilon_s) - \left(k_{d_s}\frac{fS}{\epsilon_s^2} + 2\frac{fS\dot{\epsilon}_s}{\epsilon_s^3}\right)\dot{\epsilon}_s + k_{i_s}\frac{fS}{\epsilon_s^2}\int_0^t (\epsilon_s^d - \epsilon_s)\, d\tau. \tag{71}$$

The downside to this equation is that it requires knowledge of the target size S and the focal length f. This requirement can be removed by incorporating fS into the PID gains by defining

$$\hat{k}_{p_s} \triangleq fSk_{p_s}$$
$$\hat{k}_{i_s} \triangleq fSk_{i_s}$$
$$\hat{k}_{d_s} \triangleq fSk_{d_s},$$

and by noting from Equation (69) that $\frac{fS\dot{\epsilon}_s}{\epsilon_s^2} = w$. Therefore Equation (71) becomes

$$u_z = \hat{k}_{p_s}\frac{(\epsilon_s^d - \epsilon_s)}{\epsilon_s^2} - \left(\frac{\hat{k}_{d_s}}{\epsilon_s^2} + 2\frac{w}{\epsilon_s}\right)\dot{\epsilon}_s + \frac{\hat{k}_{i_s}}{\epsilon_s^2}\int_0^t (\epsilon_s^d - \epsilon_s)\, d\tau. \tag{72}$$

7.2 Roll Attitude Hold

The equation of motion for roll is given by Eq. (38) as $\ddot{\phi} = \tau_p / J_x$. We will use a PID controller to regulate the roll attitude as

$$\tau_p = k_{p_\phi}(\phi^c - \phi) - k_{d_\phi} p + k_{i_\phi} \int_0^t (\phi^c - \phi)\, d\tau.$$

A block diagram of the control structure is shown in Figure 19.

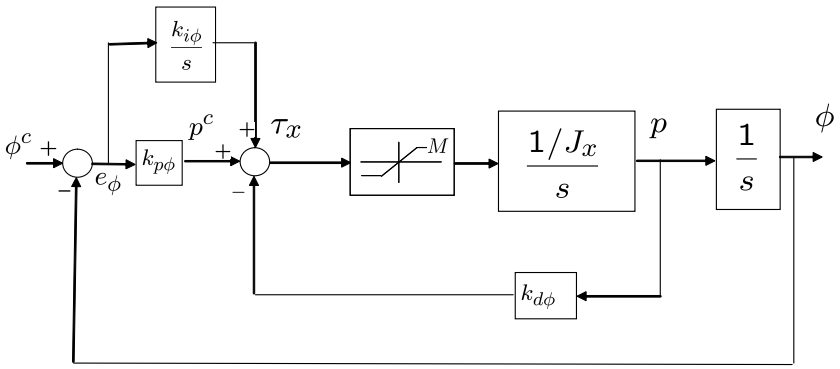

Figure 19: A block diagram of the roll attitude hold loop.

The gains k_{p_ϕ}, k_{i_ϕ}, and k_{d_ϕ} are selected one at a time using a method called successive loop closure. To pick k_{p_ϕ} note that if the input command ϕ^c is a step of value A, then at time $t = 0$, before the integrator has time to begin winding up, τ_x is given by

$$\tau_x(0) = k_{p_\phi} A.$$

Therefore, setting $k_{p_\phi} = M/A$ will just saturate the actuators when a step value of A is placed on ϕ^c.

To select k_{d_ϕ}, let k_{p_ϕ} be fixed and let $k_{i_\phi} = 0$, and solve for the characteristic equation in Evan's form verses k_{d_ϕ} to obtain

$$1 + k_{d_\phi} \frac{\frac{1}{J_x} s}{s^2 + \frac{k_{p_\phi}}{J_x}} = 0.$$

The derivative gain k_{d_ϕ} is selected to achieve a damping ratio of $\zeta = 0.9$.

The characteristic equation including k_{i_ϕ} in Evan's form verses k_{i_ϕ} is given by

$$1 + k_{i_\phi} \frac{\frac{1}{J_x}}{s^3 + \frac{k_{d_\phi}}{J_x} s^2 + \frac{k_{p_\phi}}{J_x} s} = 0.$$

The integral gain k_{i_ϕ} can be selected so that damping ratio is not significantly changed.

7.3 Pitch Attitude Hold

The equation of motion for pitch is given by Eq. (39) as $\ddot{\theta} = \tau_q / J_y$. Similar to the roll attitude hold, we will use a PID controller to regulate pitch as

$$\tau_q = k_{p_\theta}(\theta^c - \theta) - k_{d_\theta} q + k_{i_\theta} \int_0^t (\theta^c - \theta)\, d\tau.$$

7.4 Vision Based Position Tracking

From Equation 64 the lateral dynamics are given by $\ddot{p}_y = u_y$, where p_y is the relative lateral position which we drive to zero using the PID strategy

$$u_y = -k_{p_y} p_y - k_{d_y} v + k_{i_y} \int_0^t p_y\, d\tau.$$

The relative position error p_y is given by Equation (42).

From Equation 63 the longitudinal dynamics are given by $\ddot{p}_x = u_x$, where p_x is the relative lateral position which we drive to zero using the PID strategy

$$u_x = -k_{p_x} p_x - k_{d_x} u + k_{i_x} \int_0^t p_x\, d\tau.$$

The relative position error p_x is given by Equation (43).

7.5 Relative Heading Hold

The heading dynamics are given in Equation (40) as $\ddot{\psi} = \tau_r / J_z$. Define ψ^d as the inertial heading of the target, and define $\tilde{\psi} \triangleq \psi - \psi^d$ as the relative heading. The camera directly measures $\tilde{\psi}$. Assuming that ψ^d is constant we get $\ddot{\tilde{\psi}} = \tau_r / J_z$. We regulate the relative heading with the PID strategy

$$\tau_r = -k_{p_\psi} \psi - k_{d_\psi} r - k_{i_\psi} \int_0^t \psi\, d\tau.$$

7.6 Feedforward

When the quadrotor is tracking a ground robot, the motion of the robot will cause tracking errors due to the delayed response of the PID controller. If we can communicate with the robot, and we know its intended motion, we should be able to use that information to help the quadrotor predict where the robot is moving.

To motivate our approach, lets consider a simple example to gain intuition. Consider the double integrator system

$$\ddot{y} = u,$$

where y is position, and u is commanded acceleration. If r is the reference signal then the standard PD loop is shown in Figure 20.

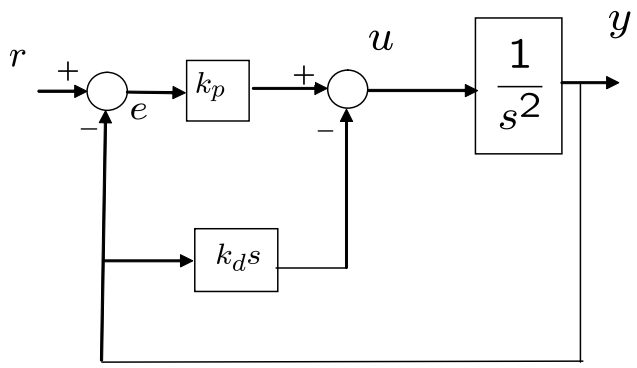

Figure 20: Standard PD loop

Let $e = r - y$, then

$$\begin{aligned}
\ddot{e} &= \ddot{r} - \ddot{y} \\
&= \ddot{r} - u \\
&= \ddot{r} - k_p e + k_d \dot{y} \\
&= \ddot{r} - k_p e - k_d \dot{e} + k_d \dot{r}.
\end{aligned}$$

In other words we have

$$\ddot{e} + k_d \dot{e} + k_p e = \ddot{r} + k_d \dot{r}.$$

Therefore, the signal $\ddot{r} + k_d \dot{r}$ drives the error transfer function $\frac{1}{s^2 + k_d s + kp}$ and if w is not zero, the tracking error will not go to zero.

From the expression $\ddot{e} = \ddot{r} - u$ we see that if instead of $u = k_p e - k_d \dot{y}$, we instead use

$$u = \ddot{r} + k_p e + k_d \dot{e},$$

then we get

$$\ddot{e} + k_d \dot{e} + k_p e = 0,$$

which ensures that $e(t) \to 0$ independent of $r(t)$. The associated block diagram is shown in Figure 21.

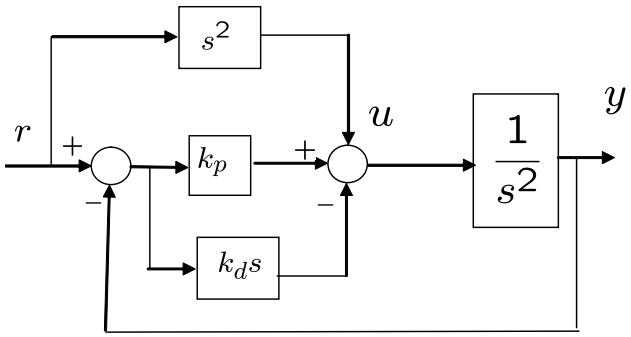

Figure 21: Feedforward term is added to a standard PD loop.

Now consider the lateral motion of the quadrotor in the cage where

$$\ddot{y} = g \tan \phi,$$

where y is the position of the quadrotor, g is the gravity constant, and ϕ is the roll angle. Let $m(t)$ be the position of the target and define $y \stackrel{\triangle}{=} m - r$ as the relative position. Then

$$\begin{aligned} \ddot{y} &= \ddot{m} - \ddot{r} \\ &= \ddot{m} - g \tan \phi. \end{aligned}$$

If by some means we know the target velocity $\dot{m}$ and the target acceleration $\ddot{m}$, then we should pick ϕ such that

$$g \tan \phi = \ddot{m} + k_p y + k_d \dot{y},$$

or

$$\phi = \tan^{-1} \left(\frac{\ddot{m} + k_p y + k_d \dot{y}}{g} \right).$$

Given the decoupling terms described at the beginning of this section, the motion model for the quadrotor in the vehicle-1 frame is given by

$$\begin{aligned}\ddot{p}_x &= u_x \\ \ddot{p}_y &= u_y \\ \ddot{\psi} &= u_\psi.\end{aligned}$$

Let the position of the target in the vehicle-1 frame be given by (m_x, m_y) and let it orientation be given by ψ_t, and define

$$\begin{aligned}\tilde{p}_x &= m_x - p_x \\ \tilde{p}_y &= m_y - p_y \\ \tilde{\psi} &= \psi_t - \psi.\end{aligned}$$

Differentiating twice we get

$$\begin{aligned}\ddot{\tilde{p}}_x &= \ddot{m}_x - u_x \\ \ddot{\tilde{p}}_y &= \ddot{m}_y - u_y \\ \ddot{\tilde{\psi}} &= \ddot{\psi}_t - u_\psi.\end{aligned}$$

Therefore, adding a feedforward term we have

$$\begin{aligned}u_x &= \ddot{m}_x + PID_x \\ u_y &= \ddot{m}_y + PID_y \\ u_\psi &= \ddot{\psi}_t + PID_\psi\end{aligned}$$

where PID_* are the PID controllers derived earlier in this chapter.

The motion model for the target is given by

$$\begin{aligned}\dot{m}_n &= V_t \cos\psi_t \\ \dot{m}_e &= V_t \sin\psi_t \\ \dot{V}_t &= u_1 \\ \dot{\psi}_t &= u_2,\end{aligned}$$

where (m_n, m_e) is the inertial position of the target, V_t is the ground speed of the target, and u_1 and u_2 are robot control signals. In the vehicle-1 frame we have

$$\begin{pmatrix}\dot{m}_x \\ \dot{m}_y\end{pmatrix} = \begin{pmatrix}\cos\psi & -\sin\psi \\ \sin\psi & \cos\psi\end{pmatrix}\begin{pmatrix}V_t\cos\psi_t \\ V_t\sin\psi_t\end{pmatrix} = \begin{pmatrix}V_t\cos\tilde{\psi} \\ V_t\sin\tilde{\psi}\end{pmatrix},$$

where the camera measures the relative heading $\tilde{\psi} = \psi - \psi_t$. Differentiating we obtain

$$\begin{pmatrix} \ddot{m}_x \\ \ddot{m}_y \end{pmatrix} = \begin{pmatrix} \cos\tilde{\psi} & \sin\tilde{\psi} \\ -\sin\tilde{\psi} & \cos\tilde{\psi} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 - \dot{\psi} \end{pmatrix}.$$

To implement feedforward on the quadrotors, we must communicate with the robot to obtain V_t, u_1, and u_2.

References

[1] B. L. Stevens and F. L. Lewis, *Aircraft Control and Simulation*. Hoboken, New Jersey: John Wiley & Sons, Inc., 2nd ed., 2003.

[2] D. Halliday and R. Resnick, *Fundamentals of Physics*. John Wiley & Sons, 3rd ed., 1988.

[3] http://www.silicondesigns.com/tech.html.

[4] M. Rauw, *FDC 1.2 - A SIMULINK Toolbox for Flight Dynamics and Control Analysis*, February 1998. Available at http://www.mathworks.com/.