# 1 Reference Frames

There are 8 reference frames formulated. Understanding the definition of each frame, and converting between these frames precisely are critical for successful flight experiments.

## 1.1 Frames Fixed to UAV

**UAV body-fixed frame:** $b$ This corresponds to the body-fixed frame defined in [2]. The center of rotor is numbered counterclockwise when observed from the top. The origin of the $b$-frame is located at the mass center of the UAV, and the first axis points toward the first rotor, and the second axis points toward the second rotor. Thus, the third axis points downward.

**IMU-fixed frame:** $i$ This corresponds to the frame marked on the surface of the IMU.

**VICON object markers frame:** $m$ When creating an object from the VICON tracker software, a reference frame fixed to the marker is defined. The orientation of this frame is identical to the above $v$-frame at the time of object creation. Therefore, it is common that the object is aligned in a specific desired way with respect to the $v$-frame.

**ZED fixed frame:** $z$ This is a frame fixed to the ZED camera. See the documents at Stereolabs website.[1] The ZED fixed frame is referred to as the camera frame in the above document, and it is right-handed with the positive Y-axis pointing down, X-axis pointing right when viewed from behind, and Z-axis pointing away from the camera.

## 1.2 Framed Fixed to Ground

**Local NED frame:** $n$ This frame is common in flight dynamics. Its origin is located at the base station. Three axes points toward the due North, the due East, and downward, respectively.

**VICON reference frame:** $v$ This corresponds to the *world* frame of the VICON system. It is determined when calibrating the VICON sensors with the marker wand. At SEH2200, there is a black electrical tape marked on the floor. The first axis points toward the right when sitting at the base station, or the 22nd street; the second axis points front when sitting at the base station, or toward the I street; the third axis points upward.

**ZED world frame:** $w$ This is the global reference frame for the ZED. If no initial parameters are set, the world Frame is located at the place where the ZED first started motion tracking, and is oriented in the direction where the device was looking.

**FDCL SEH2200 frame:** $f$ This frame is fixed to SEH2200, and its origin is identical to the origin of the $v$-frame. The first axis points front when sitting at the base station, or toward the I street; the second axis points toward the right when sitting at the base station, or the 22nd street; the third axis points downward.

## 1.3 Conversion between Frames

**Conversion among UAV-fixed frames** Conversion among the three UAV-fixed frames do not change over time, as all of them are fixed to the UAV body.

[1] https://docs.stereolabs.com/overview/positional-tracking/coordinate-frames/

It will be convenient if the IMU is aligned to the UAV body so that $R_{bi} \in \mathsf{SO}(3)$, i.e., the transform from the $i$-frame to the $b$-frame is identical, or a $\frac{\pi}{2}n$ rotation about a $b$-frame axis. However, in practice, it is challenging to align the IMU in such way. However, the IMU orientation can be precisely determined when designing the layout of the PCB board. As such, the rotation matrix $R_{bi} \in \mathsf{SO}(3)$ will be visually determined.

As discussed above, the orientation of the marker-fixed $m$-frame relative to the body-fixed $b$-frame can be determined when creating the VICON object upon the error in aligning the UAV to the black tapes on the floor.

As both of the IMU and the VICON measures attitudes, the rotation matrix $R_{mi} \in \mathsf{SO}(3)$ from the $i$-frame to the $m$-frame can be calibrated as described in Section 2.

**Conversion among ground-fixed frames** Conversion among the three ground-fixed frames do not change over time, as all of them are fixed to the ground.

The conversion between the $v$-frame and the $f$-frame is trivial, as the $f$-frame is constructed by flipping the $v$-frame. More explicitly,

$$R_{fv} = [e_2, e_1, -e_3] = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \quad R_{vf} = R_{fv}^T = R_{fv}. \tag{1}$$

The conversion between the $n$-frame and the $v$-frame is not straightforward, as the magnetic field in SEH2200 is neither uniform nor stationary. However, as both of the IMU and the VICON measures attitudes, the rotation matrix $R_{nv} \in \mathsf{SO}(3)$ from the $v$-frame to the $n$-frame can be calibrated as described in Section 2.

**Conversion between a UAV-fixed frame and a ground-fixed frame** In contrast to the two types conversion described above, the conversion between a UAV-fixed frame and a ground-fixed frame depends on the actual attitude of the UAV with respect to the ground.

Two types of measurements are available.

- $R_{ni}(t) \in \mathsf{SO}(3)$, the rotation matrix from the IMU $i$-frame to the NED $n$-frame is measured by IMU. There are several ways to convert Euler-angles to the corresponding rotation matrix. The following equations should be used for VN100.

```
YPR(0)=data->ypr.yaw*M_PI/180.;
YPR(1)=data->ypr.pitch*M_PI/180.;
YPR(2)=data->ypr.roll*M_PI/180.;

R_ni(0,0)=cos(YPR(0))*cos(YPR(1));
R_ni(0,1)=cos(YPR(0))*sin(YPR(2))*sin(YPR(1)) - cos(YPR(2))*sin(YPR(0));
R_ni(0,2)=sin(YPR(0))*sin(YPR(2)) + cos(YPR(0))*cos(YPR(2))*sin(YPR(1));
R_ni(1,0)=cos(YPR(1))*sin(YPR(0));
R_ni(1,1)=cos(YPR(0))*cos(YPR(2)) + sin(YPR(0))*sin(YPR(2))*sin(YPR(1));
R_ni(1,2)=cos(YPR(2))*sin(YPR(0))*sin(YPR(1)) - cos(YPR(0))*sin(YPR(2));
R_ni(2,0)=-sin(YPR(1));
R_ni(2,1)=cos(YPR(1))*sin(YPR(2));
R_ni(2,2)=cos(YPR(2))*cos(YPR(1));
```

- $R_{vm}(t) \in \mathsf{SO}(3)$, the rotation matrix from the marker $m$-frame to the VICON reference $v$-frame is measured by VICON. There are several ways to convert a quaternion to the corresponding rotation matrix. The following equations should be used for VICON VRPN.

```
R_vm(0,0) = 1-(2*(tdata.quat[1])*(tdata.quat[1]))-(2*(tdata.quat[2])*(tdata.quat[2]));
R_vm(0,1) = (2*tdata.quat[0]*tdata.quat[1])-(2*tdata.quat[3]*tdata.quat[2]);
R_vm(0,2) = (2*tdata.quat[0]*tdata.quat[2])+(2*tdata.quat[3]*tdata.quat[1]);
R_vm(1,0) = (2*tdata.quat[0]*tdata.quat[1])+(2*tdata.quat[3]*tdata.quat[2]);
R_vm(1,1) = 1-(2*(tdata.quat[0])*(tdata.quat[0]))-(2*(tdata.quat[2])*(tdata.quat[2]));
R_vm(1,2) = (2*(tdata.quat[1])*(tdata.quat[2]))-(2*(tdata.quat[3])*(tdata.quat[0]));
R_vm(2,0) = (2*tdata.quat[0]*tdata.quat[2])-(2*tdata.quat[3]*tdata.quat[1]);
R_vm(2,1) = (2*tdata.quat[0]*tdata.quat[3])+(2*tdata.quat[2]*tdata.quat[1]);
R_vm(2,2) = 1-(2*(tdata.quat[0])*(tdata.quat[0]))-(2*(tdata.quat[1])*(tdata.quat[1]));
```

### 1.4 Reference Frame Selection

In indoor flight experiments at SEH2200, we use the following two reference frames:

- The body-fixed frame, or the $b$-frame, as the UAV-fixed frame.
- The FDCL SEH2200 frame, or the $f$-frame, as the ground-fixed frame.

In the implementation of the controller or the generation of tracking commands,

- $x \in \mathbb{R}^3$ is the position of the origin of the $b$-frame, or the center of gravity of the UAV, resolved in the $f$-frame.
- $v = \dot{x} \in \mathbb{R}^3$.
- $R \triangleq R_{fb} \in \mathsf{SO}(3)$ is the rotation matrix from the $b$-frame to the $f$-frame.
- $\Omega \in \mathbb{R}^3$ is the angular velocity resolved in the $b$-frame.

**IMU** The output of the IMU corresponds to $(R_{\mathrm{IMU}}, \Omega_{\mathrm{IMU}}, a_{\mathrm{IMU}}) = (R_{ni}, \Omega_i, a_{\mathrm{IMU}})$, where $a_{\mathrm{IMU}}$ is the relative acceleration with respect to the gravitational acceleration resolved in the $i$-frame. They can be converted into $(R, \Omega, a)$ as

$$R(t) = R_{fb}(t) = R_{fv}R_{vn}R_{ni}(t)R_{ib}, \tag{2}$$

$$\dot{R}(t) = R_{fv}R_{vn}R_{ni}(t)\hat{\Omega}_i(t)R_{ib} = R_{fv}R_{vn}R_{ni}(t)R_{ib}\widehat{R_{ib}^T\Omega_i(t)} = R(t)\widehat{R_{ib}^T\Omega_i(t)} = R(t)\hat{\Omega}(t) \tag{3}$$

$$\Omega(t) = \Omega_b(t) = R_{bi}\Omega_i(t), \tag{4}$$

$$a(t) = \ddot{x}(t) = R_{fi}(t)a_{\mathrm{IMU}} + ge_3 = R_{fv}R_{vn}R_{ni}(t)a_{\mathrm{IMU}} + ge_3 = R_{fb}(t)R_{bi}a_{\mathrm{IMU}} + ge_3. \tag{5}$$

**VICON** The output of the VICON measurements corresponds to $(x_v, R_{vm})$. They can be converted into $(x, R)$ as

$$x(t) = x_f(t) = R_{fv}x_v(t), \tag{6}$$

$$R(t) = R_{fb}(t) = R_{fv}R_{vm}(t)R_{mb}. \tag{7}$$

**ZED** The output of the ZED measurements corresponds to $(x_w, R_{wz})$. They can be converted into $(x, R)$ as

$$x(t) = x_f(t) = R_{fw} x_w(t), \tag{8}$$

$$R(t) = R_{fb}(t) = R_{fw} R_{wz}(t) R_{zb}. \tag{9}$$

## 2 Sensor Calibration

### 2.1 IMU and VICON Calibration

The IMU measures $R_{ni} \in \mathsf{SO}(3)$ which represents the linear transformation of a representation of a vector from the $i$-frame to the $n$-frame, and the VICON measures $R_{vm} \in \mathsf{SO}(3)$ that is the rotation matrix from the $m$-frame to the $v$-frame. They are related as

$$R_{ni}(t) = R_{nv} R_{vm}(t) R_{mi}, \tag{10}$$

where $R_{mi} \in \mathsf{SO}(3)$ is the fixed rotation matrix from the $i$-frame to the $m$-frame, and $R_{nv} \in \mathsf{SO}(3)$ is the fixed rotation matrix from the $v$-frame to the $n$-frame.

The objective of this section is to compute $R_{mi}$ and $R_{nv}$ for several measurements of $\{(R_{ni}(t), R_{vm}(t)) \in \mathsf{SO}(3) \times \mathsf{SO}(3) \,|\, t \in \{t_1, t_2, \ldots t_N\}\}$ for a positive integer $N \geq 2$. This can be formulated as (11) with

$$R = R_{ni}, \quad Q = R_{vm}, \quad X = R_{nv}, \quad Y = R_{mi}.$$

The procedures to solve (11) is summarized at the end of Section 2.4.

### 2.2 VICON and ZED Calibration

For indoor flight experiments, it is desirable to calibrate the ZED measuring $R_{wz}(t)$ with the VICON measuring $R_{vm}(t)$. They are related as

$$R_{wz}(t) = R_{wv} R_{vm}(t) R_{mi} R_{ib} R_{bz},$$

which can be formulated as (11) with

$$R = R_{wz}, \quad Q = R_{vm}, \quad R_{wv} = X, \quad Y = R_{mi} R_{ib} R_{bz}.$$

The procedures to solve (11) is summarized at the end of Section 2.4. Once $Y$ is obtained, $R_{bz}$ can be constructed by

$$R_{bz} = R_{ib}^T R_{mi}^T Y.$$

### 2.3 IMU and ZED Calibration

For outdoor flight experiments, it is desirable to calibrate the ZED measuring $R_{wz}(t)$ with the IMU measuring $R_{ni}(t)$. They are related as

$$R_{wz}(t) = R_{wn} R_{ni}(t) R_{ib} R_{bz},$$

which can be formulated as (11) with

$$R = R_{wz}, \quad Q = R_{ni}, \quad R_{wn} = X, \quad Y = R_{ib} R_{bz}.$$

The procedures to solve (11) is summarized at the end of Section 2.4. Once $Y$ is obtained, $R_{bz}$ can be constructed by

$$R_{bz} = R_{ib}^T Y.$$

## 2.4 Solution of $R = XQY$

The problem of sensor calibration can be formulated as finding $X, Y \in \mathsf{SO}(3)$ for multiple pairs of measurements, $\{R_i, Q_i\}_{i=1}^N$ so that

$$R_i = XQ_iY, \tag{11}$$

for $i \in \{1, \ldots, N\}$.

For another pair of measurement indexed by $j \neq i$, we have $R_j = XQ_jY$. Since $Y = Q_j^T X^T R_j$, substituting it into the above,

$$R_i = XQ_iQ_j^T X^T R_j,$$

which is rearranged

$$R_i R_j^T = XQ_iQ_j^T X^T, \tag{12}$$

which follows the form of $A = XBX^T$, or equivalently $AX = XB$, that is an equation well-known in sensor calibrations.

We solve it using the matrix exponential. Define $r_{ij}, q_{ij} \in \mathbb{R}^3$ such that

$$R_i R_j^T = \exp \hat{r}_{ij}, \quad Q_i Q_j^T = \exp \hat{q}_{ij}. \tag{13}$$

Using this, (12) is rewritten as $\exp \hat{r}_{ij} = X \exp \hat{q}_{ij} X^T = \exp(\widehat{Xq_{ij}})$, which is equivalent to

$$r_{ij} = Xq_{ij}. \tag{14}$$

The problem of finding $X \in \mathsf{SO}(3)$ for given $\{(r_{ij}, q_{ij}) \in \mathbb{R}^3 \times \mathbb{R}^3 \,|\, i, j \in \{1, \ldots, N\}\}$ has been investigated in the field of attitude determination. In particular, the least-square determination approach is recognized as Wahba's problem.

One of the most popular solution to Wahba's problem is based on the singular value decomposition. Define the matrix $Z \in \mathbb{R}^{3\times 3}$ as

$$Z = \sum_{i,j=1}^{N} w_{ij} r_{ij} q_{ij}^T, \tag{15}$$

where $w_{ij} \in \mathbb{R}$ denotes a positive weighting parameters. Let the singular value decomposition of $Z$ be $Z = USV^T$. The least-square solution of (14) minimizing $\mathcal{J} = \sum_{i,j=1}^N w_{ij} \|r_{ij} - Xq_{ij}\|^2$ is given by

$$X = U \operatorname{diag}[1, 1, \det[U]\det[V]]\, V^T. \tag{16}$$

**Compute $X$** In summary, $X \in \mathsf{SO}(3)$ is determined by

1. Collect $\{(R_i(t), Q_i(t)) \in \mathsf{SO}(3) \times \mathsf{SO}(3) \,|\, t \in \{t_1, t_2, \ldots t_N\}\}$.
2. Define $\{(r_{ij}, q_{ij}) \in \mathbb{R}^3 \times \mathbb{R}^3 \,|\, i, j \in \{1, \ldots, N\}\}$ from (13).
3. Define $Z$ as (15), and perform singular value decomposition to obtain $Z = USV^T$.
4. We have $X = U \operatorname{diag}[1, 1, \det[U]\det[V]]\, V^T$.

**Compute $Y$** The rotation matrix $Y$ can be constructed directly by using the above solution via $Y = Q_j^T X^T R_j$ for any $j \in \{1, \dots, N\}$. In order to compute $Y$ in the lease-square sense, the above procedure can be repeated as follows.

We have $X = R_j Y^T Q_j^T$, which yield $R_i = R_j Y^T Q_j^T Q_i Y$ that is equivalent to

$$R_j^T R_i = Y^T Q_j^T Q_i Y,$$

which is comparable to (12). As such, the procedure to compute $X$ can be repeated by using the following definition of $r_{ij}, q_{ij}$ instead of (13),

$$R_j^T R_i = \exp \hat{r}_{ij}, \quad Q_j^T Q_i = \exp \hat{q}_{ij}. \tag{17}$$

In summary, the rotation matrix $Y \in \mathsf{SO(3)}$ is determined by

1. Collect $\{(R_i(t), Q_i(t)) \in \mathsf{SO(3)} \times \mathsf{SO(3)} \,|\, t \in \{t_1, t_2, \dots t_N\}\}$.
2. Define $\{(r_{ij}, q_{ij}) \in \mathbb{R}^3 \times \mathbb{R}^3 \,|\, i, j \in \{1, \dots, N\}\}$ from (17).
3. Define $Z$ as (15), and perform singular value decomposition to obtain $Z = USV^T$.
4. We have $Y = V \,\mathrm{diag}[1, 1, \det[U]\det[V]]\, U^T$.