Sensors

T. Bretl

AE 483

Fall 2021

GYROSCOPES

They measure the angular velocity:

$$w^1_{0,1} = \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} \begin{matrix} \leftarrow \text{ae483log.w\_x} \\ \leftarrow \text{ae483log.w\_y} \\ \leftarrow \text{ae483log.w\_z} \end{matrix}$$

As you will see in lab, these measurements are <u>so</u> good (they have <u>so</u> little noise) that we might as well just use them as state estimates.

So, we <u>ignore</u> these EOM:

$$\dot{w}^1_{0,1} = (J^1)^{-1}\left(\tau - \widehat{w^1_{0,1}}\, J^1 w^1_{0,1}\right) \quad \leftarrow \text{allows us to ignore } \tau$$

And we treat $w^1_{0,1}$ as an input instead of a state.

$$\begin{bmatrix} \dot{\psi} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = N w^1_{0,1}$$

Accelerometers

They measure specific force:

EOM of drone is:

$$m\ddot{z} = f_z - mg$$

① Accelerometer measures $1/m$ times all forces except gravity

$$\underbrace{\ddot{z} + g}_{\text{measurement}} = (1/m)\, f_z$$

② Accelerometer measures "acceleration without the gravity term"

$$\ddot{z} = \underbrace{\boxed{(1/m)\, f_z}}_{\text{measurement}} - g$$

EOM of proof mass is:

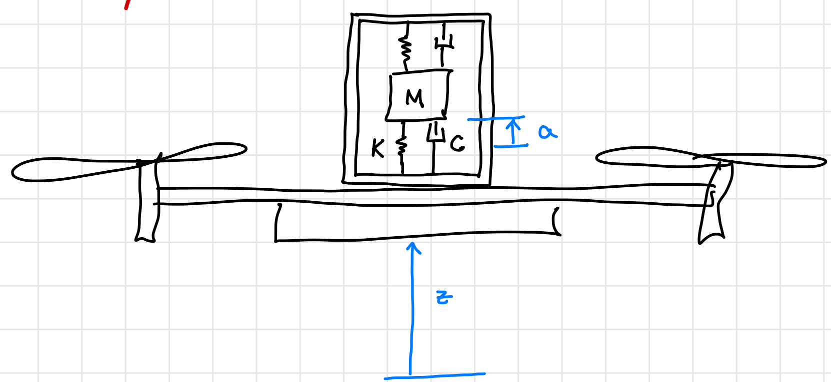
$$M(\ddot{z} + \ddot{a}) = -ka - c\dot{a} - Mg$$

If $\ddot{z}$ is constant, then steady-state is:

$$M\ddot{z} = -ka - Mg$$

The measurement is:

$$a = \underbrace{\left(-\frac{M}{K}\right)}_{\text{this is a constant that is calibrated away}} \underbrace{(\ddot{z} + g)}_{\text{this is specific force}}$$



Accelerometers

They measure specific force:

$$(R_1^0)^T \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ f_z \end{bmatrix} = m\dot{v}_{0,1}^1 + \widehat{\omega_{0,1}^1}\, m v_{0,1}^1$$

$$\underbrace{\dot{v}_{0,1}^1}_{\text{acceleration}} = (R_1^0)^T \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} + \boxed{(1/m)\begin{bmatrix} 0 \\ 0 \\ f_z \end{bmatrix} - \widehat{\omega_{0,1}^1}\, v_{0,1}^1}$$

"acceleration without the gravity term"

call this a^{SF}

This allows us to ignore f_z and replace it with a^{SF}, which we treat as an input.

ACCELEROMETERS

They measure specific force:

$$a_{SF} = (1/m)\begin{bmatrix}0\\0\\f_z\end{bmatrix} - \widehat{\omega^1_{0,1}}\, v^1_{0,1} = \begin{bmatrix}a_x\\a_y\\a_z\end{bmatrix}$$

if you want these, add code to your controller that logs sensors->acc.x and sensors->acc.y

← ae483log.a_z

$$\approx \begin{bmatrix}0\\0\\f_z/m\end{bmatrix}$$

near hover (by linearization)

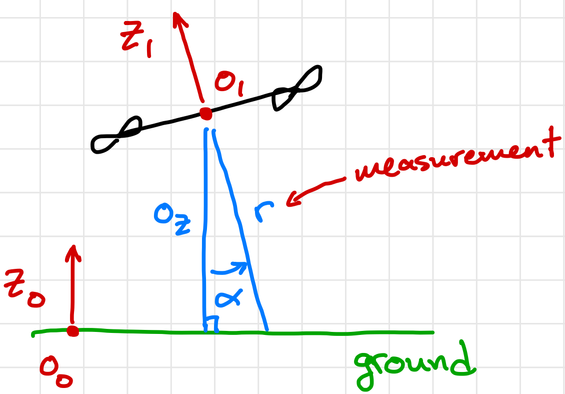
so, near hover, the x and y components of a^{SF} tell us <u>nothing</u>!

"Z RANGER"

It measures the distance along $-z_1$ from o_1 to the closest surface.

Assume that surface is "the ground" (flat, horizontal, zero height).

$$R_1^0 = \begin{bmatrix} x_1 \cdot x_0 & y_1 \cdot x_0 & z_1 \cdot x_0 \\ x_1 \cdot y_0 & y_1 \cdot y_0 & z_1 \cdot y_0 \\ x_1 \cdot z_0 & y_1 \cdot z_0 & z_1 \cdot z_0 \end{bmatrix}$$



$$\frac{o_z}{r} = \cos\alpha = \frac{z_1 \cdot z_0}{|z_1||z_0|} = z_1 \cdot z_0 = \cos\phi \cos\theta$$

$$\boxed{r = \frac{o_z}{\cos\phi \cos\theta}}$$ ← ae483log.r

$$R_1^0 = \begin{bmatrix} \cos(\psi)\cos(\theta) & \sin(\phi)\sin(\theta)\cos(\psi) - \sin(\psi)\cos(\phi) & \sin(\phi)\sin(\psi) + \sin(\theta)\cos(\phi)\cos(\psi) \\ \sin(\psi)\cos(\theta) & \sin(\phi)\sin(\psi)\sin(\theta) + \cos(\phi)\cos(\psi) & -\sin(\phi)\cos(\psi) + \sin(\psi)\sin(\theta)\cos(\phi) \\ -\sin(\theta) & \sin(\phi)\cos(\theta) & \cos(\phi)\cos(\theta) \end{bmatrix}$$

"OPTICAL FLOW"

$$n_x \approx k_{flow}\left(\frac{\overbrace{V_x - O_z\,\omega_y}^{\text{velocity of point on ground}}}{O_z}\right)$$

perspective projection

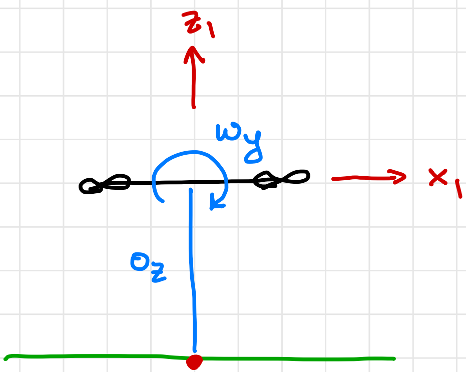
ae483log.n-x

constant parameter that depends on shutter time, pixel size, etc.

$$n_y \approx k_{flow}\left(\frac{V_y + O_z\,\omega_x}{O_z}\right)$$

ae483log.n-y

it measures the distance that points move from one image to the next, in the "x_1" and "y_1" directions



Flowdeck Measurement Model

This illustration explains how the height from the VL53L1x sensor and flow from the PMW3901 sensor are combined to calculate velocity. This has been implemented by the work of [3] and can be found in kalman_core.c in the function `kalmanCoreUpdateWithFlow()`.