

**Spaceryde Software Design Challenge Calculations****Equations of motion**X direction:

$$D = \frac{1}{2} C_D P A (V_a)^2$$

$$F_x = \frac{-D(V_x - v_w)}{V_a}, \quad a_x = \frac{-D}{m \cdot v_a} (v_x - v_w), \quad \frac{dv_x}{dt} = a_x$$

$$\frac{dv_x}{dt} + \frac{D}{m \cdot v_a} (v_x - v_w) = 0$$

$$\frac{dv_x}{ds} + \frac{D}{m \cdot v_a^2} (v_x - v_w) = 0, \quad s \leftarrow \text{arc length of trajectory, thus } \frac{ds}{dt} = v_a$$

$$\frac{dv_x}{dz} + \frac{D}{m \cdot v_a^2} (v_x - v_w) = 0, \quad dz = ds \cdot \cos(\Theta), \quad \Theta \text{ must be small, so } dz \approx ds$$

$$\frac{dv_x}{dz} + \frac{C_D P A}{2m} (v_x - v_w) = 0$$

Using initial condition of  $V_x(0) = 0$  :

$$V_x = -V_w e^{\frac{C_D P A \cdot z}{2m}} + V_w + V_{xi}, \quad V_{xi} \leftarrow \text{initial velocity of balloon, is used when } z > 1000$$

$$x = \int V_x dz = V_w e^{\frac{-C_D P A \cdot z}{2m}} * \frac{2m}{C_D P A} + V_w * z - V_w * \frac{2m}{C_D P A} + V_{xi} * z$$

$$a_x = \frac{d}{dz}(V_x) = \frac{V_w C_D P A e^{\frac{-C_D P A \cdot z}{2m}}}{2m}$$

Z direction:

$$F_y = F_{up} - mg - F_{Dz} = F_{up} - mg - \frac{1}{2} C_D P A (V_z)^2$$

$$\frac{dV_z}{dt} - \frac{F_{up} - mg - \frac{1}{2} C_D P A (V_z)^2}{m} = 0$$

Using initial condition of  $V_z(0) = 0$  :

$$V_z = \frac{\sqrt{F_{up} - mg} * \tanh\left(\frac{\sqrt{F_{up} - mg} * \sqrt{\frac{1}{2} C_D P A} * t}{m}\right)}{\sqrt{\frac{1}{2} C_D P A}}$$

$$z = \int V_z dt = \frac{m * \log(\cosh\left(\frac{\sqrt{F_{up} - mg} * \sqrt{\frac{1}{2} C_D P A} * t}{m}\right))}{\frac{1}{2} C_D P A}$$

$$a_z = \frac{d}{dz}(V_z) = \frac{(F_{up} - mg) * \text{sech}^2\left(\frac{\sqrt{F_{up} - mg} * \sqrt{\frac{1}{2} C_D P A} * t}{m}\right)}{m}$$

## Kalman filter

$A \leftarrow$  state transition matrix

$B \leftarrow$  control matrix

$U \leftarrow$  control vector

$Q \leftarrow$  process variance

$R \leftarrow$  measurement variance

$z \leftarrow$  measurement vector (raw GPS readings)

$H \leftarrow$  observation model

$w \leftarrow$  white noise vector

$X \leftarrow$  state vector

$P \leftarrow$  covariance error matrix

$K \leftarrow$  Kalman gain

Matrices to use in filter:

$$Q = \begin{pmatrix} 5^2 & 0 \\ 0 & 5^2 \end{pmatrix} R = \begin{pmatrix} 60^2 & 0 \\ 0 & 60^2 \end{pmatrix} z = \begin{pmatrix} p \\ v \end{pmatrix} H = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} P_i = \begin{pmatrix} 1000 & 0 \\ 0 & 1000 \end{pmatrix}$$

### Predict stage:

$$x_f = x_i + v\Delta t + \frac{1}{2} * a * \Delta t^2$$

$$v_f = v_i + a\Delta t$$

$$\begin{pmatrix} x \\ v \end{pmatrix} = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix} \cdot \text{preState} + \begin{pmatrix} \frac{1}{2}\Delta t^2 \\ \Delta t \end{pmatrix} \begin{pmatrix} a \end{pmatrix} + w \quad w = \begin{pmatrix} 5^2 \\ 0 \end{pmatrix}$$

state                      A                      previous state                      B                      U

Predict state:

$$\text{state} = A * \text{preState} + BU + w$$

Predict error covariance:

$$P_f = A * P_i * A^T + Q$$

### Update stage:

Get kalman gain:

$$K = P * H^T (H * P * H^T + R)^{-1}$$

Update state:

$$\text{state} = \text{prevState} + K * (z - H * \text{prevState})$$

Update error covariance:

$$P_f = P_i - K * H * P_i$$

\* Kalman filter updates at 1 hz, getting the gps info first, then waits for the accelerometer data.