<u>Spaceryde Software Design Challenge Calculations</u>

Equations of motion

<u>X direction:</u>

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

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

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

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

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

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

$\textit{Using initial condition of } V_x(0) = 0 :$

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

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

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

<u>Z direction:</u>

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

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

$\textit{Using initial condition of } V_z(0) = 0 :$

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

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

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

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:

$$Q = \begin{pmatrix} 0.1^2 & 0 \\ 0 & 0.1^2 \end{pmatrix} \quad R = \begin{pmatrix} 60^2 & 0 \\ 0 & 60^2 \end{pmatrix} \quad z = \begin{pmatrix} p \\ v \end{pmatrix} \quad H = \begin{pmatrix} 0 & 1 \end{pmatrix} \quad P_i = \begin{pmatrix} 1000 & 0 \\ 0 & 1000 \end{pmatrix}$$

<u>Predict stage:</u>

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

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

state *A* *previous state* *B* *U*

Predict state:

$state = A * preState + BU + w$

Predict error covariance:

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

<u>Update stage:</u>

Get kalman gain:

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

Update state:

$state = prevState + K * (z - H * prevState)$

Update error covariance:

$P_f = (I - K * H) * P_i$