

1 Quaternion Multiplication

$$\begin{aligned}Q_{W\text{new}} &= Q_{W1}(Q_{W2}) - Q_{X1}(Q_{X2}) - Q_{Y1}(Q_{Y2}) - Q_{Z1}(Q_{Z2}) \\Q_{X\text{new}} &= Q_{W1}(Q_{X2}) - Q_{X1}(Q_{W2}) - Q_{Y1}(Q_{Z2}) - Q_{Z1}(Q_{Y2}) \\Q_{Y\text{new}} &= Q_{W1}(Q_{Y2}) - Q_{X1}(Q_{Z2}) - Q_{Y1}(Q_{W2}) - Q_{Z1}(Q_{X2}) \\Q_{Z\text{new}} &= Q_{W1}(Q_{Z2}) - Q_{X1}(Q_{Y2}) - Q_{Y1}(Q_{X2}) - Q_{Z1}(Q_{W2})\end{aligned}$$

2 Quaternion Division

$$\begin{aligned}Q_{W\text{new}} &= Q_{W1}(Q_{W2}) + Q_{X1}(Q_{X2}) + Q_{Y1}(Q_{Y2}) + Q_{Z1}(Q_{Z2}) \\Q_{X\text{new}} &= -Q_{W1}(Q_{X2}) + Q_{X1}(Q_{W2}) + Q_{Y1}(Q_{Z2}) - Q_{Z1}(Q_{Y2}) \\Q_{Y\text{new}} &= -Q_{W1}(Q_{Y2}) - Q_{X1}(Q_{Z2}) + Q_{Y1}(Q_{W2}) + Q_{Z1}(Q_{X2}) \\Q_{Z\text{new}} &= -Q_{W1}(Q_{Z2}) + Q_{X1}(Q_{Y2}) - Q_{Y1}(Q_{X2}) + Q_{Z1}(Q_{W2})\end{aligned}$$

3 Quaternion Conjugate

$$\begin{aligned}Q_{W\text{conjugate}} &= Q_{W\text{input}} \\Q_{X\text{conjugate}} &= -Q_{X\text{input}} \\Q_{Y\text{conjugate}} &= -Q_{Y\text{input}} \\Q_{Z\text{conjugate}} &= -Q_{Z\text{input}}\end{aligned}$$

4 Quaternion Normal

$$Normal_{\text{rational}} = Q_W^2 + Q_X^2 + Q_Y^2 + Q_Z^2$$

5 Quaternion Multiplicative Inverse

$$Q_{\text{reciprocal}} = Q_{\text{conjugate}} \frac{1}{Q_{\text{normal}}}$$

6 Quaternion Vector Rotation

$$\begin{aligned} Q_{\text{Wbivector}} &= 0 \\ Q_{\text{Xbivector}} &= V_X \\ Q_{\text{Ybivector}} &= V_Y \\ Q_{\text{Zbivector}} &= V_Z \end{aligned}$$

$$Q_{\text{rotation}} = Q_{\text{current}}(Q_{\text{bivector}})(Q_{\text{reciprocal}})$$

$$\begin{aligned} V_{\text{Xrotated}} &= Q_{\text{Xrotation}} \\ V_{\text{Yrotated}} &= Q_{\text{Yrotation}} \\ V_{\text{Zrotated}} &= Q_{\text{Zrotation}} \end{aligned}$$

7 Quaternion Vector Rotation Removal

$$\begin{aligned} Q_{\text{Wbivector}} &= 0 \\ Q_{\text{Xbivector}} &= V_X \\ Q_{\text{Ybivector}} &= V_Y \\ Q_{\text{Zbivector}} &= V_Z \end{aligned}$$

$$Q_{\text{rotation}} = Q_{\text{conjugate}}(Q_{\text{bivector}})(Q_{\text{reciprocal}})$$

$$\begin{aligned} V_{\text{Xrotated}} &= Q_{\text{Xrotation}} \\ V_{\text{Yrotated}} &= Q_{\text{Yrotation}} \\ V_{\text{Zrotated}} &= Q_{\text{Zrotation}} \end{aligned}$$

8 Unit Quaternion

$$\begin{aligned} Q_{\text{Wunit}} &= \frac{Q_{\text{Winput}}}{Q_{\text{normal}}} \\ Q_{\text{Xunit}} &= \frac{Q_{\text{Xinput}}}{Q_{\text{normal}}} \\ Q_{\text{Yunit}} &= \frac{Q_{\text{Yinput}}}{Q_{\text{normal}}} \\ Q_{\text{Zunit}} &= \frac{Q_{\text{Zinput}}}{Q_{\text{normal}}} \end{aligned}$$

9 Quaternion Dot Product

$$D_{\text{dot}} = Q_{\text{W1}}(Q_{\text{W2}}) + Q_{\text{X1}}(Q_{\text{X2}}) + Q_{\text{Y1}}(Q_{\text{Y2}}) + Q_{\text{Z1}}(Q_{\text{Z2}})$$

10 Quaternion Magnitude

$$M_{\text{magnitude}} = \sqrt{Q_{\text{normal}}}$$

11 Quaternion Additive Inverse

$$\begin{aligned} Q_{\text{Wnegative}} &= -Q_{\text{Winput}} \\ Q_{\text{Xnegative}} &= -Q_{\text{Xinput}} \\ Q_{\text{Ynegative}} &= -Q_{\text{Yinput}} \\ Q_{\text{Znegative}} &= -Q_{\text{Zinput}} \end{aligned}$$

12 Quaternion Smooth Interpolation Between Quaternions

$$\begin{aligned} Q_{\text{initial}} &= Q_{\text{Unit initial}} \\ Q_{\text{final}} &= Q_{\text{Unit final}} \end{aligned}$$

$$D_{\text{dot}} = Q_{\text{initial}} \cdot Q_{\text{final}}$$

$$Q_{\text{initial}} = \begin{cases} Q_{\text{initial}} = Q_{\text{initial}(\text{additive inverse})}, & \text{if } D_{\text{dot}} < 0. \\ Q_{\text{initial}}, & \text{otherwise.} \end{cases}$$

$$D_{\text{dot}} = |D_{\text{dot}}|$$

$$D_{\text{dot}} = \begin{cases} 1, & \text{if } D_{\text{dot}} > 1. \\ D_{\text{dot}}, & \text{otherwise.} \end{cases}$$

$$\theta = \arccos(D_{\text{dot}}) \times \text{ratio}$$

$$\begin{aligned} Q_{\text{orthonormal}} &= Q_{\text{final}} - Q_{\text{initial}} \times D_{\text{dot}} \\ Q_{\text{output}} &= Q_{\text{initial}} \times \cos(\theta) + Q_{\text{orthonormal}} \times \sin(\theta) \end{aligned}$$

13 Quadcopter Combined Thrust Vector

$$Q_{\text{change}} = \left(\frac{2 \times (Q_{\text{target}} - Q_{\text{current}}) \times Q_{\text{current conjugate}}}{dT} \right)$$

$$V_{X\text{change}} = Q_{X\text{change}}$$

$$V_{Y\text{change}} = Q_{Y\text{change}}$$

$$V_{Z\text{change}} = Q_{Z\text{change}}$$

$$V_{\text{RotationOutput}} = \text{FeedbackController}_{\text{rotation}}.\text{Calculate}(0, V_{\text{change}})$$

$$V_{\text{PositionOutput}} = \text{FeedbackController}_{\text{position}}.\text{Calculate}(0, V_{\text{CurrentPosition}} - V_{\text{TargetPosition}})$$

$$V_{Y\text{ThrusterBOutput}} = -V_{X\text{RotationOutput}} + V_{Z\text{RotationOutput}} - V_{Y\text{RotationOutput}}$$

$$V_{Y\text{ThrusterCOutput}} = -V_{X\text{RotationOutput}} - V_{Z\text{RotationOutput}} + V_{Y\text{RotationOutput}}$$

$$V_{Y\text{ThrusterDOutput}} = V_{X\text{RotationOutput}} - V_{Z\text{RotationOutput}} - V_{Y\text{RotationOutput}}$$

$$V_{Y\text{ThrusterEOutput}} = V_{X\text{RotationOutput}} + V_{Z\text{RotationOutput}} + V_{Y\text{RotationOutput}}$$

$$V_{\text{HoverAngles}} = \text{RotationToHoverAngles}(Q_{\text{CurrentRotation}})$$

$$V_{\text{PositionOutput}} = \text{CalculateRotationOffset}(Q_{\text{CurrentRotation}}).\text{RotateVector}(V_{\text{PositionOutput}})$$

$$V_{X\text{PositionOutput}} = V_{X\text{PositionOutput}} + V_{Z\text{HoverAngles}}$$

$$V_{Z\text{PositionOutput}} = V_{Z\text{PositionOutput}} - V_{X\text{HoverAngles}}$$

$$V_{\text{ThrusterBOutput}} = V_{\text{ThrusterBOutput}} + V_{\text{PositionOutput}}$$

$$V_{\text{ThrusterCOutput}} = V_{\text{ThrusterCOutput}} + V_{\text{PositionOutput}}$$

$$V_{\text{ThrusterDOutput}} = V_{\text{ThrusterDOutput}} + V_{\text{PositionOutput}}$$

$$V_{\text{ThrusterEOutput}} = V_{\text{ThrusterEOutput}} + V_{\text{PositionOutput}}$$

14 Quadcopter Thruster Position Calculation

$$V_{\text{ThrusterBPosition}} = Q_{\text{CurrentRotation}}.\text{RotateVector}(V_{\text{ThrusterBOffset}}) + V_{\text{TargetPosition}}$$

$$V_{\text{ThrusterCPosition}} = Q_{\text{CurrentRotation}}.\text{RotateVector}(V_{\text{ThrusterCOffset}}) + V_{\text{TargetPosition}}$$

$$V_{\text{ThrusterDPosition}} = Q_{\text{CurrentRotation}}.\text{RotateVector}(V_{\text{ThrusterDOffset}}) + V_{\text{TargetPosition}}$$

$$V_{\text{ThrusterEPosition}} = Q_{\text{CurrentRotation}}.\text{RotateVector}(V_{\text{ThrusterEOffset}}) + V_{\text{TargetPosition}}$$

15 Quadcopter Hover Angle Calculation

$$DA_{\text{Direction}} = \text{RotationMatrix}.\text{RotateVector}(EA_{\text{rotate}}(0, -90, 0), DA_{\text{Direction}})$$

$$DA_{\text{Direction}} = \text{RotationMatrix}.\text{RotateVector}(EA_{\text{rotate}}(0, DA_{\text{Rotation}}, 0), DA_{\text{Direction}})$$

$$D_{\text{InnerJoint}} = \text{RadiansToDegrees}(\arcsin(D_{\text{DirectionVectorZ}}))$$

$$D_{\text{OuterJoint}} = \text{RadiansToDegrees}(\arctan2(D_{\text{DirectionVectorX}}, D_{\text{DirectionVectorY}}))$$

16 Quadcopter Estimate Position

$$V_{TBThrust} = Vector(0, ThrustBOutputY, 0)$$

$$V_{TCThrust} = Vector(0, ThrustCOutputY, 0)$$

$$V_{TDThrust} = Vector(0, ThrustDOutputY, 0)$$

$$V_{TEThrust} = Vector(0, ThrustEOutputY, 0)$$

$$Q_{TBR} = EA(ThrustBOutput.X, 0, -ThrustBOutput.Z)$$

$$Q_{TCR} = EA(ThrustCOutput.X, 0, -ThrustCOutput.Z)$$

$$Q_{TDR} = EA(ThrustDOutput.X, 0, -ThrustDOutput.Z)$$

$$Q_{TER} = EA(ThrustEOutput.X, 0, -ThrustEOutput.Z)$$

$$V_{TBThrust} = Q_{TBR}.RotateVector(TBThrust)$$

$$V_{TCThrust} = Q_{TCR}.RotateVector(TCThrust)$$

$$V_{TDThrust} = Q_{TDR}.RotateVector(TDThrust)$$

$$V_{TEThrust} = Q_{TER}.RotateVector(TEThrust)$$

$$V_{ThrustSum} = V_{TBThrust} + V_{TCThrust} + V_{TDThrust} + V_{TEThrust}$$

$$V_{ThrustSum} = Q_{current}.RotateVector(V_{ThrustSum})$$

$$V_{XDragForce} = D_{AirDensity} \times D_{XCurrentVelocity}^2 \times D_{DragCoefficient} \times Sign(XCurrentVelocity)$$

$$V_{YDragForce} = D_{AirDensity} \times D_{YCurrentVelocity}^2 \times D_{DragCoefficient} \times Sign(YCurrentVelocity)$$

$$V_{ZDragForce} = D_{AirDensity} \times D_{ZCurrentVelocity}^2 \times D_{DragCoefficient} \times Sign(ZCurrentVelocity)$$

$$V_{CurrentAcceleration} = V_{ThrustSum} + V_{WorldAcceleration}$$

$$V_{CurrentVelocity} = V_{CurrentVelocity} + V_{CurrentAcceleration} \times D_{TimeDerivative}$$

$$V_{CurrentPosition} = V_{CurrentPosition} + V_{CurrentVelocity} \times D_{TimeDerivative}$$

17 Quadcopter Estimate Rotation

$$V_{TB} = V_{TB}ThrustVector$$

$$V_{TC} = V_{TC}ThrustVector$$

$$V_{TD} = V_{TD}ThrustVector$$

$$V_{TE} = V_{TE}ThrustVector$$

$$V_{TB} = Q_{CurrentRotation} \cdot RotateVector(TB)$$

$$V_{TC} = Q_{CurrentRotation} \cdot RotateVector(TC)$$

$$V_{TD} = Q_{CurrentRotation} \cdot RotateVector(TD)$$

$$V_{TE} = Q_{CurrentRotation} \cdot RotateVector(TE)$$

$$D_{Torque} = D_{ArmLength} \times \sin(180 - D_{ArmAngle})$$

$$V_{XAngularAcceleration} = (V_{TBY} + V_{TCY} - V_{TDY} - V_{TEY}) \times D_{Torque}$$

$$V_{YAngularAcceleration} = (V_{TBX} + V_{TCX} - V_{TDX} - V_{TEX}) \times D_{Torque}$$

$$V_{ZAngularAcceleration} = (V_{TBZ} + V_{TCZ} - V_{TDZ} - V_{TEZ}) \times D_{Torque}$$

$$V_{ZAngularAcceleration} = (-V_{TBY} + V_{TCY} + V_{TDY} - V_{TEY}) \times D_{Torque}$$

$$V_{XDragForce} = D_{AirDensity} \times D_{XAngularVelocity}^2 \times D_{DragCoefficient} \times Sign(XAngularVelocity)$$

$$V_{YDragForce} = D_{AirDensity} \times D_{YAngularVelocity}^2 \times D_{DragCoefficient} \times Sign(YAngularVelocity)$$

$$V_{ZDragForce} = D_{AirDensity} \times D_{ZAngularVelocity}^2 \times D_{DragCoefficient} \times Sign(ZAngularVelocity)$$

$$V_{DifferentialThrust} = V_{TB} + V_{TC} - V_{TD} - V_{TE}$$

$$V_{AngularAcceleration} = V_{AngularAcceleration} + V_{DifferentialThrust}$$

$$V_{AngularVelocity} = V_{AngularVelocity} + (V_{AngularAcceleration} - V_{DragForce}) \times D_{TimeDerivative}$$

$$Q_{AngularRotation} = \frac{V_{AngularVelocity} \times D_{TimeDerivative}}{2}$$

$$Q_{AngularPosition} = Q_{AngularPosition} + Q_{AngularRotation} \times Q_{AngularPosition}$$

18 Quadcopter Calculate 3D Yaw

$$V_{HoverRotation} = HoverAnglesFromQuaternion(Q_{CurrentRotation})$$

$$Q_{Hover} = EA(V_{HoverRotation})$$

$$Q_{Yaw3D} = Q_{Hover} \times Q_{CurrentRotation} \cdot MultiplicativeInverse()$$