RotorHead

March 11, 2024

Contents

0.0.1 Introduction

I wanted to learn more about how a helicopter works, specifically how the rotors are controlled. So I built a rotor-head out of LEGO and did some math. I also relearned a lot about typesetting equations I knew about a decade ago. Finally, learned about a method of helicopter control known as cyclic-collective pitch mixing (CCPM), which is the algorithm I'll be discussing here.

0.0.2 Definitions

Helicopter

An aircraft that uses airflow over a rotating wing to generate lift, rather than using thrust to push a fixed wing through the air as in a traditional airplane. A generic helicopter has a main rotor that generates life, and a tail rotor that counteracts the yaw induced by the main rotor torque on the fuselage and provides directional control. The main rotor of is attached to the fuselage by an assembly known as the rotor-head. The drive shaft for the main rotor emerges from the transmission through the top of the fuselage.

The primary controls of a helicopter are a stick, called the cyclic, a lever, called the collective, and a pair of pedals, called anti-torque pedals. The throttle is typically managed mechanically to keep the rotor RPM constant. The cyclic stick controls roll and pitch of the helicopter. The anti-torque pedals control the thrust of the tail rotor. The collective changes the angle of attack of the main rotor, to regulate the thrust of the rotor, but can really be thought of as the "gas pedal" of the helicopter.

Swashplate

A swashplate is a device that allows a rotating element to be inclined with two degrees of freedom, pitch and roll, and also move vertically. The bottom half is attached to three linear actuators, which provide all degrees of freedom. The rotating top half is linked to the drive shaft through the "scissor-link". Push rods connected to the top of the swashplate vary the pitch of the rotors based on the cyclic and collective inputs.

Cyclic-Collective Pitch Mixing

An algorithm for calculating fly-by-wire helicopter control, based on the control of radio controlled helicopters. Three servos are driven from inputs mixed from the collective and cyclic inputs in such a way that collective and cyclic appear to be independent variables

Collective

Helicopter control which collectively adjusts the angle of attack of the rotor blades, affecting overall rotor thrust. Controlled by a lever.

Cyclic

Helicopter control which affects the deflection of the rotor disk, based on pitch and roll. Controlled by a joystick.

0.0.3 Coordinate systems

- P0 is origin of coordinate system, centered under the mast of the helicopter.
- Positive X is pointing to the front of the helicopter.
- Positive Y is pointing port.
- Positive Z is pointing up.

see Fig 1

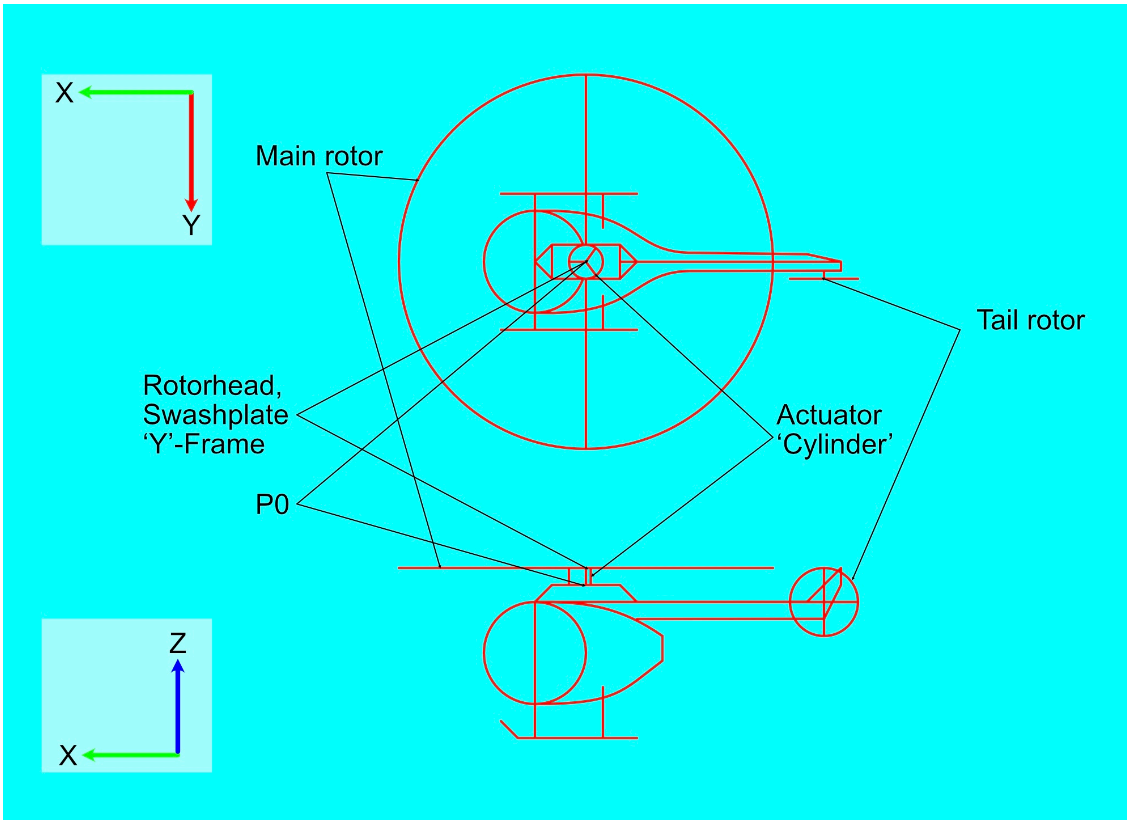


Fig 1: Basic diagram of a helicopter and coordinate system

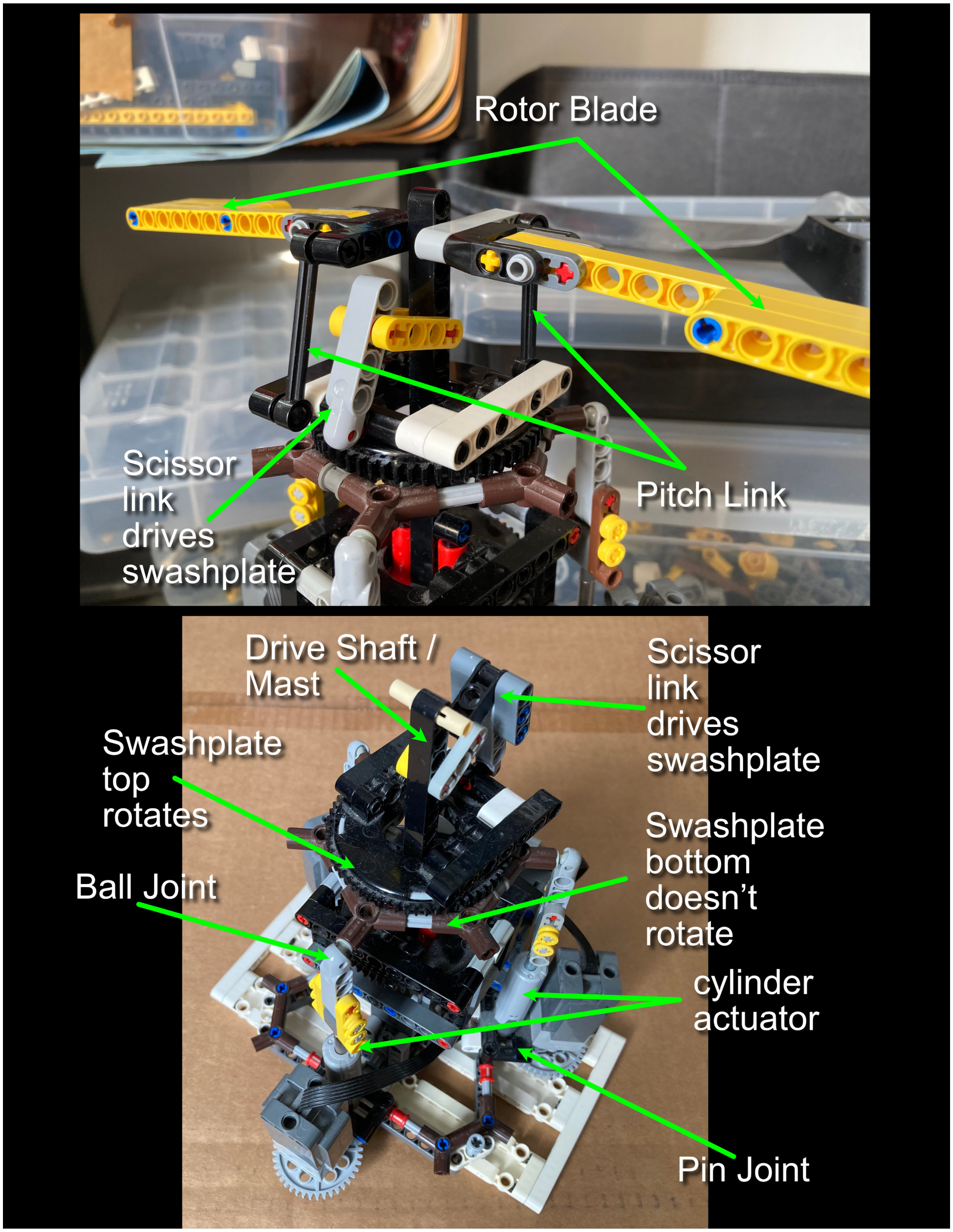


Fig 2: Model rotor head with and without rotors attached

«eq.1» $$\vec{V_p} = \begin{bmatrix}\cos\left(\angle A_p\right) & 0 & \sin\left(\angle A_p\right)\end{bmatrix}$$

«eq.2» $$\hat{V_p} = \frac{\vec{V_p}}{\left|\vec{V_p}\right|}$$

«eq.3» $$\vec{V_r} = \begin{bmatrix}0 & \cos\left(\angle A_r\right) & \sin\left(\angle A_r\right)\end{bmatrix}$$

«eq.4» $$\hat{V_r} = \frac{\vec{V_r}}{\left|\vec{V_r}\right|}$$

«eq.5» $$\vec{F_f} = \begin{bmatrix}R_{sw} & 0 & 0\end{bmatrix}$$

«eq.6» $$\vec{F_p} = \begin{bmatrix}R_{sw}\sin\left(\angle A_{cyl}\right) & R_{sw}\cos\left(\angle A_{cyl}\right) & 0\end{bmatrix}$$

«eq.7» $$\vec{F_s} = \begin{bmatrix}R_{sw}\sin\left(2\angle A_{cyl}\right) & R_{sw}\cos\left(2\angle A_{cyl}\right) & 0\end{bmatrix}$$

«eq.8» $$\vec{V_{disk}} = \hat{V_p} \times \hat{V_r}$$

Then, for every Cylinder $\vec{C_N}$

«eq.9» $$\vec{V_{mast}} = \begin{bmatrix}0 & 0 & P_{coll}\end{bmatrix}$$

«eq.10» $$\vec{F_N} = P_{fn} - P_O$$

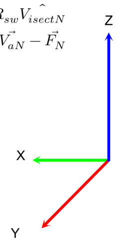
«eq.11» $$\vec{V_{cN}} = \vec{F_N} \times \vec{V_{mast}}$$

«eq.12» $$\vec{V_{isectN}} = -\vec{V_{cN}} \times \vec{V_{disk}}$$

«eq.13» $$\hat{V_{isectN}} = \frac{\vec{V_{isectN}}}{\left|\vec{V_{isectN}}\right|}$$

«eq.14» $$\vec{V_{aN}} = \vec{V_{mast}} + R_{sw}\hat{V_{isectN}}$$

«eq.15» $$\vec{C_N} = \vec{V_{aN}} - \vec{F_N}$$



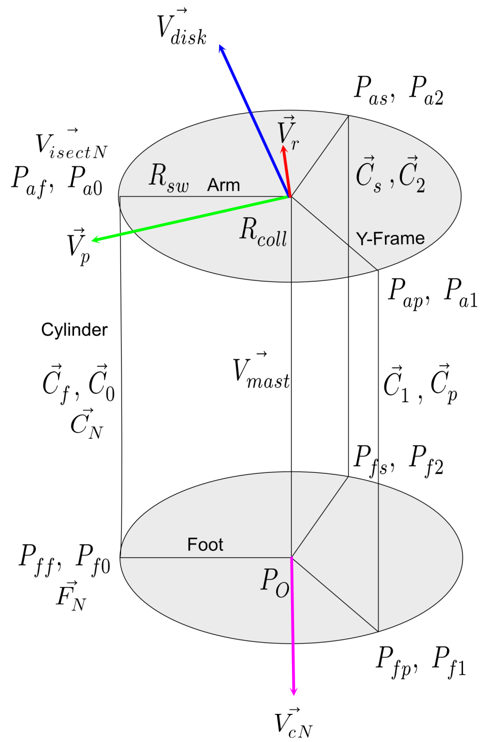


Fig 3: System diagram and equations

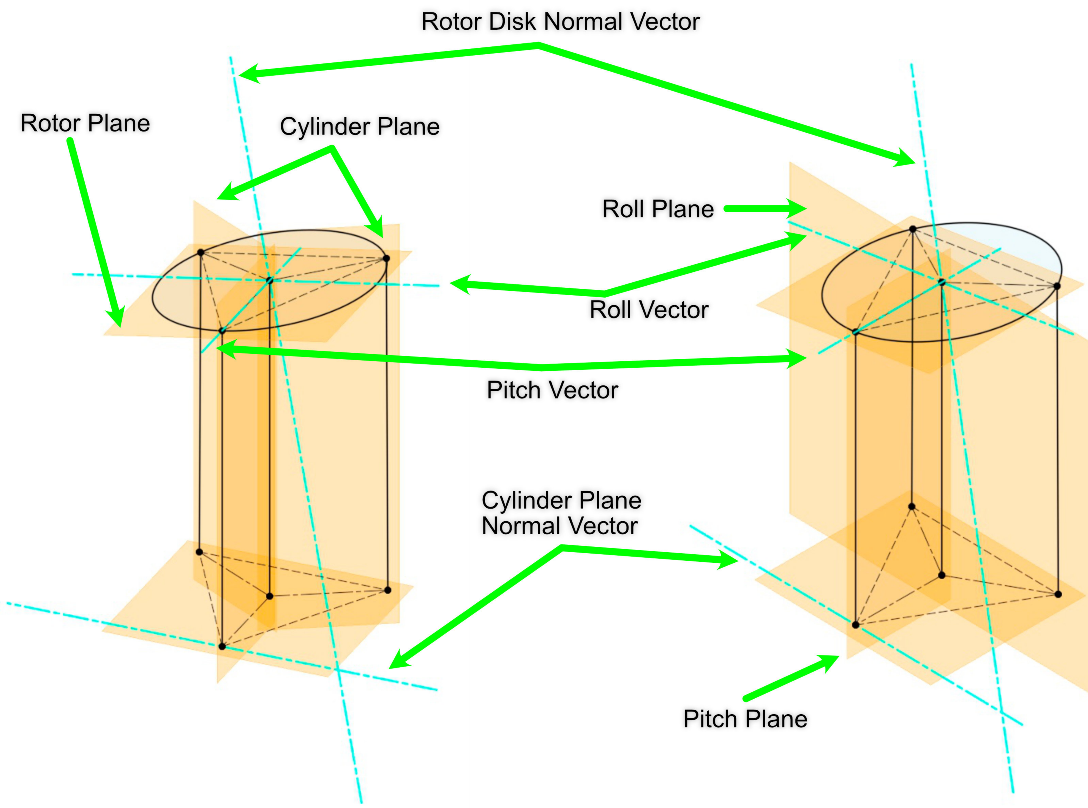


Fig 4: Important vectors and planes

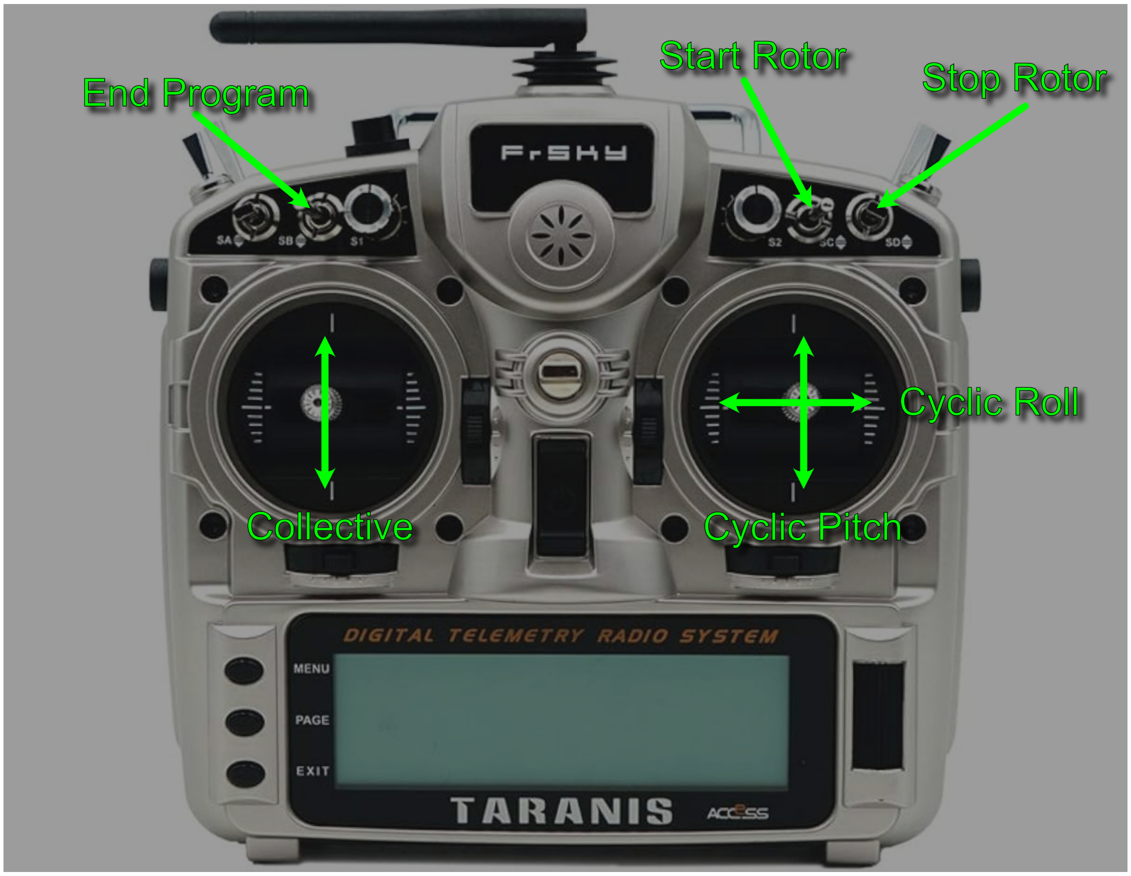


Fig 5: Control Scheme

0.0.4 System Description

Our rotorhead system (Fig 2), constructed from LEGO Technic bricks and programmed with an alternative Control+ firmware called PyBricks, consists of a two-part rotating swashplate, the bottom half of which is represented metaphorically as a Y-shaped frame (Fig 3) rotating around a spherical bearing, which in turn surrounds a bushing through which the drive shaft slides.

(Note that our model system doesn't actually have this bearing, it's all faked with math, because I don't have a part that will do the job. Kind of the whole point of this exercise. I don't wanna break my bricks.)

The major vectors and planes of the system are shown in Fig 4.

This allows the swashplate to pitch and roll around the spherical bearing, which in turn moves up and down on the drive shaft.
The arms of the notional Y-shaped frame will be referred to as swashpate arms, each with a length of R_{sw}.

The center of the Y-frame is referred to as P_{coll}, and represents the hight of the rotorhead over the Origin.
P_{coll} can vary in Z-height regulated by the collective position.
The ends of the arms are referred to as $\vec{V_{aN}}$ with N proceeding counter-clockwise from the front position.

The arms are connected to the helicopter with linear actuators, which will be referred to as cylinders. The cylinders are connected to the ends of the arms with ball joints, and to the base XY plane by pins on "feet", which have an identical Y-shape to the swashplate frame.

Thus, the cylinders can rotate towards the origin along their feet.
The length of these feet is the same as the length of the arms R_{sw}, but needn't be. The cylinders are located at 120 intervals around a circle centered on the Origin, but don't need to be regularly distributed.

The ends of the feet are $\vec{F_N}$.

Cylinders are numbered, but are also referred to by their positions, front, port, and starboard for convenience.

Each cylinder has a minimum and maximum length, C_{min} and C_{max}.

The cyclic input is used to make pitch and roll vectors $\vec{V_p}$ and $\vec{V_r}$, and collective input the vector $\vec{V_{mast}}$.

The control is provided by a radio control transmitter connected to the base computer via USB as shown in Fig 5.

0.0.5 System Equations

$$
\begin{aligned}
&«eq.1» & \vec{V_p} &= \begin{bmatrix}\cos\left(\angle A_p\right) & 0 & \sin\left(\angle A_p\right)\end{bmatrix} \\
&«eq.2» & \hat{V_p} &= \frac{\vec{V_p}}{\left|\vec{V_p}\right|} \\
&«eq.3» & \vec{V_r} &= \begin{bmatrix}0 & \cos\left(\angle A_r\right) & \sin\left(\angle A_r\right)\end{bmatrix} \\
&«eq.4» & \hat{V_r} &= \frac{\vec{V_r}}{\left|\vec{V_r}\right|} \\
&«eq.5» & \vec{F_f} &= \begin{bmatrix}R_{sw} & 0 & 0\end{bmatrix} \\
&«eq.6» & \vec{F_p} &= \begin{bmatrix}R_{sw}\sin\left(\angle A_{cyl}\right) & R_{sw}\cos\left(\angle A_{cyl}\right) & 0\end{bmatrix} \\
&«eq.7» & \vec{F_s} &= \begin{bmatrix}R_{sw}\sin\left(2\angle A_{cyl}\right) & R_{sw}\cos\left(2\angle A_{cyl}\right) & 0\end{bmatrix} \\
&«eq.8» & \vec{V_{disk}} &= \hat{V_p} \times \hat{V_r}
\end{aligned}
\tag{1}
$$

Then, for every Cylinder $\vec{C_N}$

$$
\begin{aligned}
&«eq.9» & \vec{V_{mast}} &= \begin{bmatrix}0 & 0 & P_{coll}\end{bmatrix} \\
&«eq.10» & \vec{F_N} &= P_{fn} - P_O \\
&«eq.11» & \vec{V_{cN}} &= \vec{F_N} \times \vec{V_{mast}} \\
&«eq.12» & \vec{V_{isectN}} &= -\vec{V_{cN}} \times \vec{V_{disk}} \\
&«eq.13» & \hat{V_{isectN}} &= \frac{\vec{V_{isectN}}}{\left|\vec{V_{isectN}}\right|} \\
&«eq.14» & \vec{V_{aN}} &= \vec{V_{mast}} + R_{sw}\hat{V_{isectN}} \\
&«eq.15» & \vec{C_N} &= \vec{V_{aN}} - \vec{F_N}
\end{aligned}
\tag{2}
$$

Plane P_α

Plane P_β

$\vec{V_{\alpha 1}}$

$\vec{V_{\alpha 2}}$

$\vec{V_{\beta 1}}$

$\vec{V_{\beta 2}}$

Normal Vector $\vec{V_{\alpha N}}$

Normal Vector $\vec{V_{\beta N}}$

Solution Vector $\vec{V_{\alpha \beta N}}$

0.0.6 Cyclic-Collective Pitch Mixing (CCPM) Algorithm

The pitch vector $\vec{V_p}$ is the x axis rotated around the y axis by the pitch input.
The roll vector $\vec{V_r}$ is the y axis rotated along the x axis by the roll input.

The normal vector $\vec{V_{disk}}$ representing the plane of the rotor disk is then $\vec{V_p}$ X $\vec{V_r}$ [eq. 8].

The point P_{coll} is chosen from the collective input, yielding $\vec{V_{mast}}$ [eq. 9].

Then, for each of the cylinders $\vec{C_N}$, the foot vector $\vec{F_N}$ is the the line between the origin P_O and the base of the cylinder P_{fn}.

The cylinder plane vector $\vec{V_{cN}}$ is then $\vec{F_N}$ X $\vec{V_{mast}}$ [eq. 11].

The cross product of $\vec{V_{disk}}$ and $\vec{V_{cN}}$ is the intersection of the rotor disk and the cylinder plane, which is our answer, normalized to $\hat{V_{isectN}}$ [eq. 12].

$\hat{V_{isectN}}$ is multiplied by R_{sw} and added to $\vec{V_{mast}}$, yielding the arm vector $\vec{V_{aN}}$ [eq. 14].

The cylinder vector $\vec{C_N}$ is then $\vec{V_{aN}}$ - $\vec{F_N}$ [eq. 15].

Finally, the length of the cylinder vector is checked to make sure it fits within the limits.

0.0.7 Source Code of CCPM Solve Function

```python
def solve(self, pitch, roll, collpct):
    "Implements collective-cyclic pitch mixing algorithm"
    Vp = lin.vector(m.cos(m.radians(pitch)), 0, m.sin(m.radians(pitch)))
    Vr = lin.vector(0, m.cos(m.radians(roll)), m.sin(m.radians(roll)))

    # Normal of rotor disk
    Vdisk = lin.cross(Vp, Vr)
    Vdisk_n = lin.normalize(Vdisk)

    # top of mast at collective setting
    Pcoll = self.Cmin + collpct*self.Crange
    Vmast = lin.vector(0, 0, Pcoll)
    arms = []
    for i, Fn in enumerate(self.feet):
        # Vcn is the plane the cylinder rotates on its foot in, Foot X Mast
        Vcn = lin.cross(Fn, Vmast)
        Vcn_n = lin.normalize(Vcn)

        # Visect is the intersection of the Rotor Disk plane and the Cylinder rotation plane
        Visect = lin.cross(Vdisk_n, Vcn_n) # should be plane intersection
        Visect_n = lin.normalize(Visect)

        # Va is the arm vector as rotated in the cylinder rotation plane
        Va = (self.Rsw * Visect_n) + Vmast

        arms.append(Va)
        cyl_len = lin.vmag(Va - Fn)
        if cyl_len < self.Cmin:
            raise ValueError(f"too short! Cyl: {cyl_len:{4}.{4}} min: {self.Cmin:{4}}")
        elif cyl_len > self.Cmax:
            raise ValueError(f"too long! Cyl: {cyl_len:{4}.{4}} max: {self.Cmax:{4}}")

    (Cf, Cp, Cs) = arms


    #successfully validated, save old values and calculate cylinder lengths
    self.s_Cf = Cf
    self.s_Cp = Cp
    self.s_Cs = Cs
    self.s_Vmast = Vmast

    #cylinder lengths in mm
    self.cfc = lin.vmag(Cf - self.Ff)
    self.cpc = lin.vmag(Cp - self.Fp)
    self.csc = lin.vmag(Cs - self.Fs)
    return (Cf, Cp, Cs, Vmast)
```

0.0.8 Description of source code

Main algorithm - swashplate.py: the code that does the actual swash plate calculation

Executable programs, two of which run on the host computer and one on the Technic Hub. - rotormath.py: an animated simulation of the algorithm - rotorbase.py: base station that decodes gamepad data and sends it to the hub - rotor.py: the control program that runs on the Technic Hub

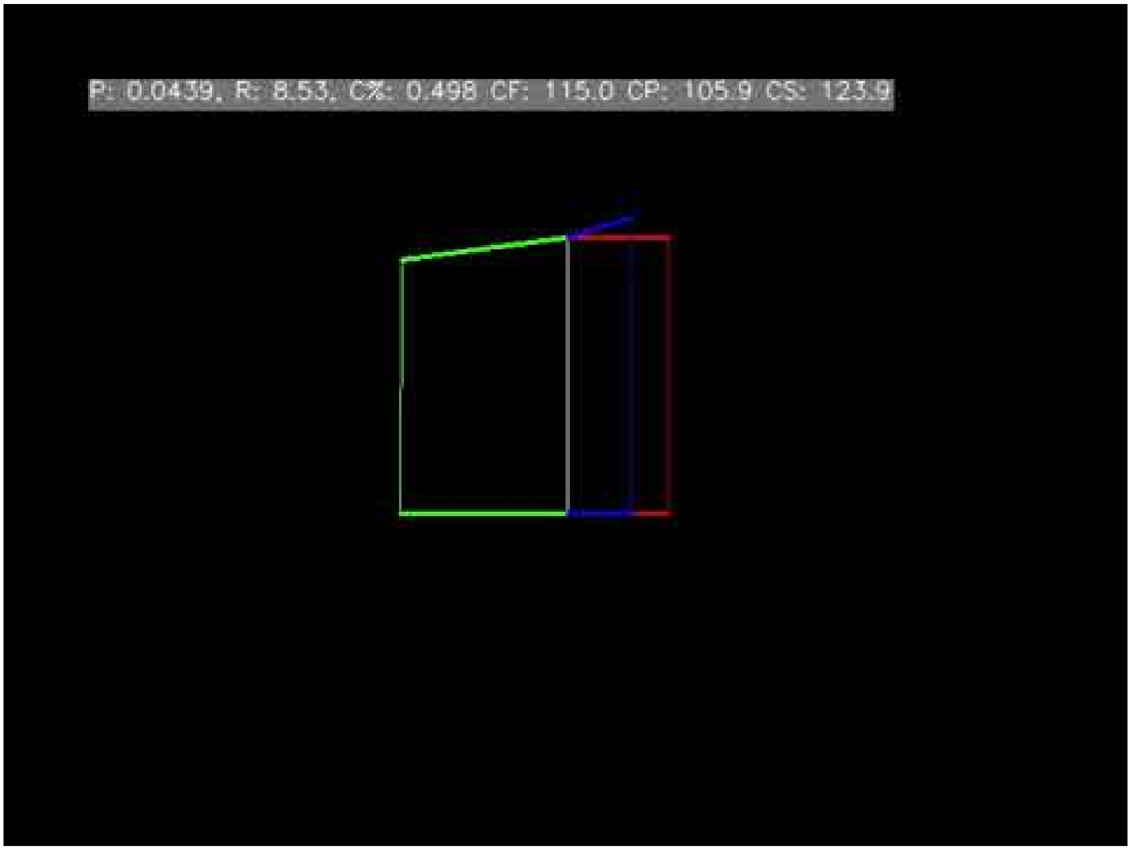
Library code
- linear.py: a translation layer to make linear algebra identical under numpy and pybricks - controllers.py: game controller interface - joycode.py: a lightweight wire protocol for transmitting game controller data to the hub - cvgraph.py: opencv-based animation convenience functions

0.0.9 Video Demonstrations

'Demonstration of Cyclic-Collective Pitch Mixing and controls'

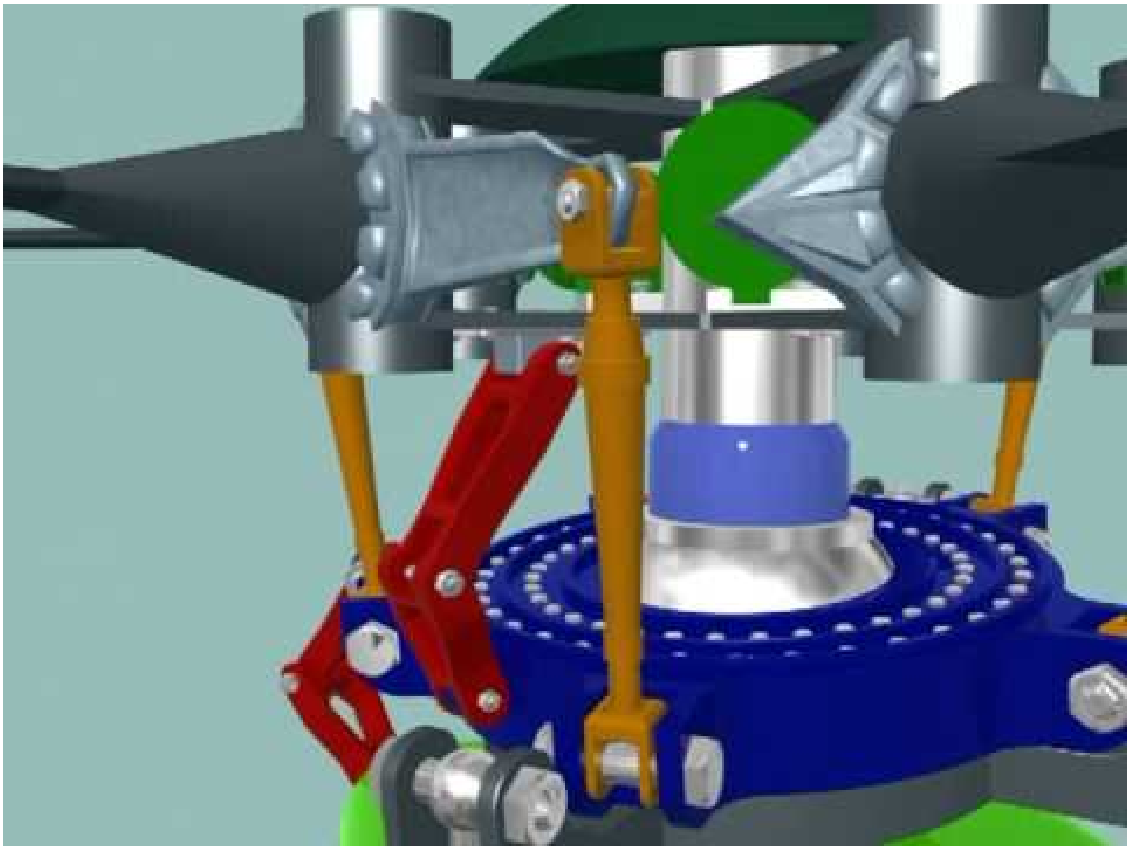
'Rotorhead side-on in better light'

'Rotormath simulation animation'



0.0.10 References

This animation of an S-61 Sea King rotor head was invaluable in my understanding of how a helicopter works.



- What Is CCPM, on an RC Helicopter & Why it's Important?
- Cyclic/collective pitch mixing (wikipedia)
- PyBricks Main Site