Understanding Trigonometry for a Circular Knob Control

This guide explains how trigonometry is used to lay out and program a circular knob, like those found in music synthesizers or audio interfaces.

1. The Circle Equation
   Any point on a circle centered at (cx, cy) with radius r can be found with:
      x = cx + r × cos(θ)
      y = cy + r × sin(θ)
   Here, θ (theta) is the angle in radians.

2. Visualizing the Angles
   - 0° points right (3 o'clock)
   - 90° points down (6 o'clock)
   - 180° points left (9 o'clock)
   - 270° points up (12 o'clock)
   SwiftUI's coordinate system flips Y, so angles appear clockwise.

3. Mapping Value to Angle
   A knob's value (0 to 1) can map to a range of angles, for example:
      startAngle = 210° (7 o'clock)
      endAngle = 330° (5 o'clock)
      angle = startAngle + (value × 300°)

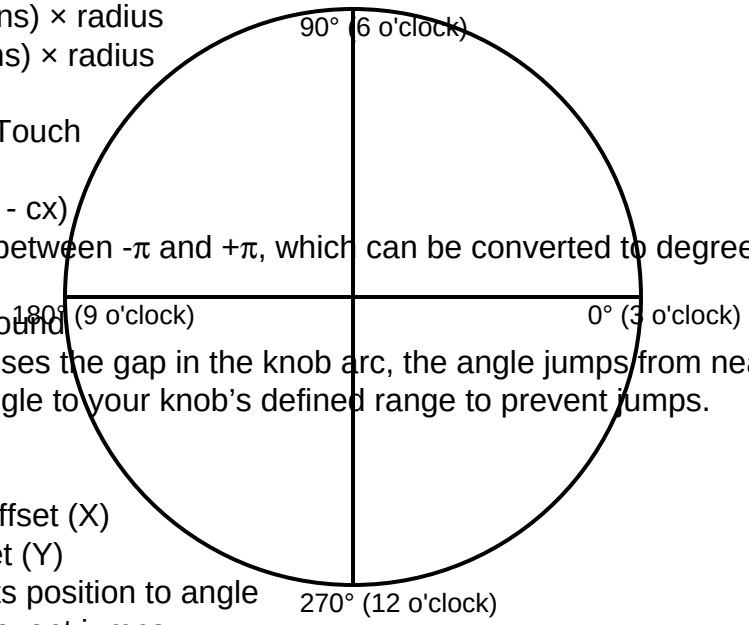
4. Converting Degrees to Radians
   Computers use radians:
      radians = degrees × π / 180

5. Getting X and Y
   Once you have radians, position the indicator:
      x = cx + cos(radians) × radius
      y = cy + sin(radians) × radius

6. Getting Angle from Touch
   When a user drags:
      θ = atan2(y - cy, x - cx)
   This gives an angle between -π and +π, which can be converted to degrees.

7. Preventing Wrap-around
   When the touch crosses the gap in the knob arc, the angle jumps from near 360° to 0°.
   Clamp or limit the angle to your knob's defined range to prevent jumps.

8. Summary
   - cos(θ): horizontal offset (X)
   - sin(θ): vertical offset (Y)
   - atan2(y, x): converts position to angle
   - Clamp values to prevent jumps



Trigonometry of a Circular Knob

Think of trigonometry as a way to describe movement around a circle — you're converting between "angles" and "positions."