Antikythera Rings

Technical Documentation & Simulation Specification

1 Project Overview

The **Antikythera Rings** project is a real-time astronomical simulation inspired by the ancient Antikythera mechanism, which was discovered near the Greek island of Antikythera in the Aegean Sea.

The simulation dynamically tracks and displays the positions of the seven main celestial bodies visible to the naked human eye:

- The Sun
- The Moon
- Mercury
- Venus
- Mars
- Jupiter
- Saturn

1.1 Timezone and Localization Support

To facilitate local observations, users can select any timezone to view celestial positions synchronized to their local time. For example, a user in Chicago can select the `UTC-6` timezone (Central Standard Time). For enhanced user convenience, the interface provides secondary identifiers (such as `-05 CDT` or `-06 CST` for Central Time zones depending on Daylight Saving Time) and references three major cities per timezone to guide selection.

2 Astronomical Coordinate System

To mathematically define and map the locations of these celestial bodies, astronomers utilize various coordinate systems. The Antikythera Rings model primarily implements the **Ecliptic Coordinate System**, which describes positions based on the angular components along the ecliptic plane.

2.1 The Ecliptic Line

The *ecliptic line* represents the apparent path that the Sun appears to travel across the celestial sphere over the course of a full year, as observed from Earth.

2.2 Coordinate Representation

Ecliptic coordinates are expressed using two primary dimensions:

1. **Ecliptic Longitude (λ):** Measured along the ecliptic plane, starting at the Vernal Equinox (the point where the ecliptic intersects the celestial equator). It ranges continuously from 0° to 360°.
2. **Ecliptic Latitude (β):** Specifies the angular distance north or south of the ecliptic plane, measured perpendicularly to the plane.

2.3 Computational Backend

The simulation utilizes the **Astronomy Engine** library developed by Don Cross (available on GitHub). Within the system, the `geo_vec` function is called to compute and extract the geocentric ecliptic coordinates, which directly dictate the precise rotational displacement required for each concentric ring of the mechanism.

3 Mathematical & Technical Implementation

In the rendering pipeline, the exact coordinate center of the display canvas is mapped to the coordinate variables $(\text{centerX}, \text{centerY})$, establishing the mathematical origin $(0, 0)$. Each concentric ring undergoes an angular displacement (θ) directly derived from its respective celestial body's ecliptic longitude (λ).

3.1 Ring Rotation Synchronization

The rotation and alignment of the simulation rings are implemented across multiple platforms:

3.2 JavaScript Implementation

In the HTML5 Canvas version, the coordinate context rotation is synchronized as follows:

```
ctx.rotate((-angles[index] * Math.PI) / 180);
```

3.3 Python / Kivy Implementation

In the Python desktop application framework, the transformation is applied via:

```
Rotate(angle=-angle, origin=(0,0))
```

3.4 Time Scales and Epochs

High-precision orbital tracking requires a uniform time standard. The Astronomy Engine automatically converts standard Universal Time Coordinated (`UTC`) timestamps into **Ephemeris Time** (or Terrestrial Time, `TT`). This conversion is crucial because `UTC` is fundamentally tied to the Earth's non-uniform rotation, whereas planetary synchronization demands an invariant, regular time-scale.

Once computed, this uniform time is referenced against the **J2000.0 epoch**, which is precisely calibrated to start on 1 January 2000 at 12:00 TT. The resulting epoch offset is used to solve the planetary orbital elements and plot the precise angular positions on the ecliptic rings. The online version of the tool provides a live readout of the ecliptic longitude in degrees for all tracked bodies.

4 References and Further Reading

For additional information regarding the underlying coordinate frameworks and rotation physics, please consult the following resources:

- Equatorial Coordinate System:
  `https://en.wikipedia.org/wiki/Equatorial_coordinate_system`
- Astronomical Coordinate Systems:
  `https://en.wikipedia.org/wiki/Astronomical_coordinate_systems`
- Fixed-Axis Rotation Dynamics:
  `https://en.wikipedia.org/wiki/Rotation_around_a_fixed_axis`
- Horizontal Coordinate System:
  `https://en.wikipedia.org/wiki/Horizontal_coordinate_system`