Tracking Rates Integration Guide

Quick Reference

Tracking Rates (degrees/second)

Rate	Value (°/s)	Relative to Sidereal	Use For
Sidereal	0.00417807	1.000×	Stars, DSOs
Solar	0.00416667	0.997×	Sun tracking
Lunar	0.00402667	0.964×	Moon tracking
King	0.00418952	1.003×	Circumpolar objects

Movement Rates (as sidereal multiples)

Rate	Multiple	Degrees/Second	Use For
Very Slow	0.25×	0.00104	Very fine control
Slow	0.5×	0.00209	Fine control
Guide	1×	0.00418	**Guiding/tracking**
Fast Guide	2×	0.00836	Fast corrections
Center	4×	0.01671	**Centering objects**
Find	8×	0.03342	**Finding objects**
Move	16×	0.06685	Moving to targets
Slew	24×	0.10027	**Standard slewing**
Fast Slew	40×	0.16712	High-speed slewing
Max	60×	0.25068	Maximum speed

Integration Steps

Step 1: Add Constants to telescope.py

At the top of your `telescope.py`, after imports:

python

```python
# ============================================================================
# Tracking Rate Constants
# ============================================================================

# Base tracking rates (degrees/second)
SIDEREAL_RATE = 0.0041780746  # Stars
SOLAR_RATE = 0.0041666667     # Sun
LUNAR_RATE = 0.0040266670     # Moon
KING_RATE = 0.0041895210      # Circumpolar

# Sidereal multipliers for manual control
class SiderealMultiplier:
    VERY_SLOW = 0.25
    SLOW = 0.5
    GUIDE = 1.0
    FAST_GUIDE = 2.0
    CENTER = 4.0
    FIND = 8.0
    MOVE = 16.0
    SLEW = 24.0
    FAST_SLEW = 40.0
    MAX = 60.0
```

Step 2: Add Conversion Methods to Telescope Class

Add these methods to your `OnStepXTelescope` class:

python

```python
class OnStepXTelescope:
    # ... existing __init__ and methods ...

    # ========================================================================
    # Tracking Rate Helper Methods
    # ========================================================================

    def move_axis_sidereal_rate(self, axis, sidereal_multiple):
        """
        Move axis at sidereal rate multiple

        Args:
            axis: TelescopeAxes.axisPrimary or axisSecondary
            sidereal_multiple: Multiple of sidereal rate
                               Positive for East/North
                               Negative for West/South
                               Zero to stop

        Examples:
            # Guide rate (1× sidereal)
            telescope.move_axis_sidereal_rate(axis, 1.0)

            # Center rate (8× sidereal)
            telescope.move_axis_sidereal_rate(axis, 8.0)

            # Slew west (24× sidereal)
            telescope.move_axis_sidereal_rate(axis, -24.0)
        """
        rate_deg_per_sec = sidereal_multiple * SIDEREAL_RATE
        self.move_axis(axis, rate_deg_per_sec)

    def move_axis_solar_rate(self, axis, solar_multiple=1.0):
        """
        Move axis at solar rate (for Sun tracking)

        Args:
            axis: Telescope axis
            solar_multiple: Multiple of solar rate (default 1.0)
        """
        rate_deg_per_sec = solar_multiple * SOLAR_RATE
        self.move_axis(axis, rate_deg_per_sec)

    def move_axis_lunar_rate(self, axis, lunar_multiple=1.0):
        """
        Move axis at lunar rate (for Moon tracking)

        Args:
```

```
        axis: Telescope axis
        lunar_multiple: Multiple of lunar rate (default 1.0)
    """
    rate_deg_per_sec = lunar_multiple * LUNAR_RATE
    self.move_axis(axis, rate_deg_per_sec)

def move_axis_king_rate(self, axis, king_multiple=1.0):
    """
    Move axis at King rate (for circumpolar objects)

    Args:
        axis: Telescope axis
        king_multiple: Multiple of King rate (default 1.0)
    """
    rate_deg_per_sec = king_multiple * KING_RATE
    self.move_axis(axis, rate_deg_per_sec)
```

Usage Examples

Example 1: Basic Tracking Rates

```python
from telescope import TelescopeAxes

# Track stars at sidereal rate
telescope.move_axis_sidereal_rate(TelescopeAxes.axisPrimary, 1.0)

# Track the Sun at solar rate
telescope.move_axis_solar_rate(TelescopeAxes.axisPrimary, 1.0)

# Track the Moon at lunar rate
telescope.move_axis_lunar_rate(TelescopeAxes.axisPrimary, 1.0)

# Track circumpolar object at King rate
telescope.move_axis_king_rate(TelescopeAxes.axisPrimary, 1.0)

# Stop
telescope.move_axis_sidereal_rate(TelescopeAxes.axisPrimary, 0)
```

Example 2: Manual Control at Standard Rates

python

```python
from telescope import SiderealMultiplier, TelescopeAxes

# Center an object (8× sidereal)
telescope.move_axis_sidereal_rate(
    TelescopeAxes.axisPrimary,
    SiderealMultiplier.CENTER
)

# Find an object quickly (16× sidereal)
telescope.move_axis_sidereal_rate(
    TelescopeAxes.axisPrimary,
    SiderealMultiplier.MOVE
)

# Slew to target (24× sidereal)
telescope.move_axis_sidereal_rate(
    TelescopeAxes.axisPrimary,
    SiderealMultiplier.SLEW
)

# Guide at slow rate (0.5× sidereal)
telescope.move_axis_sidereal_rate(
    TelescopeAxes.axisPrimary,
    SiderealMultiplier.SLOW
)
```

Example 3: Direction Control

python

```python
# Move East at center rate
telescope.move_axis_sidereal_rate(TelescopeAxes.axisPrimary, 8.0)

# Move West at center rate (negative)
telescope.move_axis_sidereal_rate(TelescopeAxes.axisPrimary, -8.0)

# Move North at guide rate
telescope.move_axis_sidereal_rate(TelescopeAxes.axisSecondary, 1.0)

# Move South at guide rate (negative)
telescope.move_axis_sidereal_rate(TelescopeAxes.axisSecondary, -1.0)
```

Example 4: Solar System Object Tracking

```python
# Track the Sun (Solar rate, both axes if needed)
telescope.move_axis_solar_rate(TelescopeAxes.axisPrimary, 1.0)

# Track the Moon (Lunar rate)
telescope.move_axis_lunar_rate(TelescopeAxes.axisPrimary, 1.0)

# Track a planet (use sidereal, since planets move slowly)
# For precise planetary tracking, you'd adjust rates dynamically
telescope.move_axis_sidereal_rate(TelescopeAxes.axisPrimary, 1.0)
```

Example 5: Satellite Tracking (Direct Rates)

```python
# For satellites, use direct degrees/second
# (Not sidereal multiples, since satellites move much faster)

# ISS overhead pass example
ra_rate = 0.5    # degrees/second eastward
dec_rate = 0.3   # degrees/second northward

telescope.move_axis(TelescopeAxes.axisPrimary, ra_rate)
telescope.move_axis(TelescopeAxes.axisSecondary, dec_rate)

# Update rates every second as satellite moves
# (handled by tracking software)
```

Example 6: Combined Axes Movement

```python
# Center object moving both axes simultaneously
telescope.move_axis_sidereal_rate(TelescopeAxes.axisPrimary, 8.0)   # East
telescope.move_axis_sidereal_rate(TelescopeAxes.axisSecondary, 4.0) # North

time.sleep(2)  # Move for 2 seconds

# Stop both axes
telescope.move_axis_sidereal_rate(TelescopeAxes.axisPrimary, 0)
telescope.move_axis_sidereal_rate(TelescopeAxes.axisSecondary, 0)
```

When to Use Each Rate

Sidereal Rate

- **Default tracking rate**
- For stars and deep-sky objects
- Most common use case
- RA tracking for equatorial mounts

Solar Rate

- Tracking the Sun (with proper filters!)
- Solar observations
- Slightly slower than sidereal (0.997×)
- Compensates for Earth's orbit

Lunar Rate

- Tracking the Moon
- Lunar photography
- Slower than sidereal (0.964×)
- Compensates for Moon's orbital motion

King Rate

- For circumpolar objects
- Objects very close to celestial pole
- Slightly faster than sidereal (1.003×)
- Rarely used in practice

Sidereal Multiples

- Manual telescope control
- Object centering and framing
- Use higher multiples for faster movement
- Guide rate (1×) for tracking corrections

Testing Your Rates

Quick Test Script

python

```python
#!/usr/bin/env python3
"""Test tracking rates"""

from telescope import OnStepXTelescope, TelescopeAxes
from telescope import SIDEREAL_RATE, SOLAR_RATE, LUNAR_RATE, KING_RATE
from telescope import SiderealMultiplier

# Connect to telescope
telescope = OnStepXTelescope(
    connection_type='network',
    host='192.168.1.100'
)
telescope.connect()

print("\n" + "="*60)
print("Testing Tracking Rates")
print("="*60)

# Test each tracking rate
rates_to_test = [
    ("Sidereal", 1.0, SIDEREAL_RATE),
    ("Solar", 1.0, SOLAR_RATE),
    ("Lunar", 1.0, LUNAR_RATE),
    ("King", 1.0, KING_RATE),
]

for name, multiplier, rate in rates_to_test:
    print(f"\nTesting {name} rate ({rate:.9f} °/s)...")
    telescope.move_axis(TelescopeAxes.axisPrimary, rate * multiplier)
    time.sleep(3)
    telescope.move_axis(TelescopeAxes.axisPrimary, 0)
    print(f"  ✓ {name} rate works")

# Test sidereal multipliers
print("\n" + "="*60)
print("Testing Sidereal Multipliers")
print("="*60)

multipliers_to_test = [
    ("Guide", SiderealMultiplier.GUIDE),
    ("Center", SiderealMultiplier.CENTER),
    ("Slew", SiderealMultiplier.SLEW),
]

for name, mult in multipliers_to_test:
    rate = mult * SIDEREAL_RATE
    print(f"\nTesting {name} ({mult}× sidereal = {rate:.6f} °/s)...")
```

```
    telescope.move_axis_sidereal_rate(TelescopeAxes.axisPrimary, mult)
    time.sleep(2)
    telescope.move_axis_sidereal_rate(TelescopeAxes.axisPrimary, 0)
    print(f"  ✓ {name} rate works")

print("\n" + "="*60)
print("All rates tested successfully!")
print("="*60 + "\n")

telescope.disconnect()
```

Rate Comparison Table

Run this to see all rates:

```
python

from telescope import SIDEREAL_RATE, SOLAR_RATE, LUNAR_RATE, KING_RATE

print("\nTRACKING RATES COMPARISON")
print("="*70)
print(f"{'Rate':<12} {'°/second':<15} {'\"/second':<15} {'Relative':<15}")
print("-"*70)

rates = [
    ("Sidereal", SIDEREAL_RATE, 1.000),
    ("Solar", SOLAR_RATE, SOLAR_RATE/SIDEREAL_RATE),
    ("Lunar", LUNAR_RATE, LUNAR_RATE/SIDEREAL_RATE),
    ("King", KING_RATE, KING_RATE/SIDEREAL_RATE),
]

for name, rate, relative in rates:
    arcsec = rate * 3600
    print(f"{name:<12} {rate:<15.9f} {arcsec:<15.6f} {relative:.5f}×")

print("="*70 + "\n")
```

Common Mistakes to Avoid

❌ Wrong: Using rate selector commands

```
python

# DON'T DO THIS - old incorrect implementation
telescope.send_command(':RG#')  # This is just "set to guide rate"
telescope.send_command(':Me#')
```

✅ Right: Using variable rate commands

```python
# DO THIS - correct implementation
rate = SIDEREAL_RATE * 8.0  # 8× sidereal
telescope.move_axis(TelescopeAxes.axisPrimary, rate)
```

❌ Wrong: Confusing tracking rate with movement rate

```python
# Lunar rate is NOT a movement speed
# It's a tracking rate for the Moon
telescope.move_axis_lunar_rate(axis, 8.0)  # This doesn't make sense
```

✅ Right: Use appropriate rate types

```python
# For tracking the Moon
telescope.move_axis_lunar_rate(axis, 1.0)

# For moving telescope quickly
telescope.move_axis_sidereal_rate(axis, 8.0)
```

Summary

What You Want	Use This Method	Example
Track stars	`move_axis_sidereal_rate()`	`(axis, 1.0)`
Track Sun	`move_axis_solar_rate()`	`(axis, 1.0)`
Track Moon	`move_axis_lunar_rate()`	`(axis, 1.0)`
Track circumpolar	`move_axis_king_rate()`	`(axis, 1.0)`
Center object	`move_axis_sidereal_rate()`	`(axis, 8.0)`
Find object	`move_axis_sidereal_rate()`	`(axis, 16.0)`
Slew quickly	`move_axis_sidereal_rate()`	`(axis, 24.0)`
Satellite tracking	`move_axis()`	`(axis, 0.5)`

All methods ultimately call `move_axis(axis, degrees_per_second)` which is the ASCOM-standard implementation!