SpaceTux Industries

KSP Wheel Modules

A complete reference to all the KSP Wheel modules

Linuxgurugamer
12-18-2025

Contents

KSP Wheel Modules Complete Reference

This is the single consolidated reference for Kerbal Space Program (KSP) 1.12.5 wheel-related PartModules. It includes module explanations, field descriptions, runtime behavior notes, example ModuleManager configuration blocks, tuning recipes, troubleshooting decision tables, presets, and direct mappings to the WheelTunerWindow UI.

ModuleWheelBase

Purpose:
Defines the core wheel collider and physical properties shared by all wheel modules.

Common Fields:

- radius (float) – Wheel radius
- mass (float) – Wheel mass contribution
- frictionMultiplier (float) – Global friction scaling

Runtime Notes:

- Changes often require a wheel rebuild
- Affects all other wheel modules

Example ModuleManager CFG:
```
@MODULE[ModuleWheelBase]
{
  @radius = 0.45
  @mass = 0.05
  @frictionMultiplier = 1.2
}
```

ModuleWheelSuspension

Purpose:
Controls suspension travel, spring force, damping, and anti-roll behavior.

Common Fields:

- suspensionDistance (float)
- springRatio (float)
- damperRatio (float)
- targetPosition (float)
- antiRoll (float)

Runtime Notes:

- Stiff springs cause bounce
- Low damper causes oscillation

Example ModuleManager CFG:
```
@MODULE[ModuleWheelSuspension]
{
  @suspensionDistance = 0.3
  @springRatio = 8
  @damperRatio = 1.5
}
```

ModuleWheelSteering

Purpose:
Provides steering capability with optional speed-based curves.

Common Fields:

- steeringEnabled (bool)
- steeringResponse (float)
- steeringCurve (FloatCurve)

Runtime Notes:

- High-speed steering should be limited
- Curve edits require rebuild

Example ModuleManager CFG:
```
@MODULE[ModuleWheelSteering]
{
  @steeringEnabled = true
  @steeringResponse = 5
  !steeringCurve {}
  %steeringCurve
  {
    key = 0 30 0 0
    key = 20 10 -1 -1
  }
}
```

ModuleWheelMotor

Purpose:
Applies motor torque to powered wheels.

Common Fields:

- motorEnabled (bool)
- maxTorque (float)
- torqueCurve (FloatCurve)

Runtime Notes:

- High torque can flip light rovers
- Torque curve defines speed envelope

Example ModuleManager CFG:
```
@MODULE[ModuleWheelMotor]
{
 @motorEnabled = true
 @maxTorque = 20
 !torqueCurve {}
 %torqueCurve
 {
  key = 0 20 0 0
  key = 25 5 -1 -1
 }
}
```

ModuleWheelBrakes

Purpose:
Controls braking strength and response.

Common Fields:

- brakeEnabled (bool)
- maxBrakeTorque (float)
- brakeResponse (float)

Runtime Notes:

- Too much torque causes skidding

Example ModuleManager CFG:
```
@MODULE[ModuleWheelBrakes]
{
 @maxBrakeTorque = 150
 @brakeResponse = 2
}
```

ModuleWheelDeployment

Purpose:
Handles deployable landing gear animation and state.

Common Fields:

- retractable (bool)

- deploySpeed (float)
- deployedPosition (float)
- retractedPosition (float)

Runtime Notes:

- Animation mismatches cause visual issues

Example ModuleManager CFG:
```
@MODULE[ModuleWheelDeployment]
{
 @deploySpeed = 2
 @deployedPosition = 1
 @retractedPosition = 0
}
```

ModuleWheelDamage

Purpose:
Defines damage tolerance and repair behavior.

Common Fields:

- stressTolerance (float)
- impactTolerance (float)
- repairable (bool)

Runtime Notes:

- Very high tolerances reduce realism

Example ModuleManager CFG:
```
@MODULE[ModuleWheelDamage]
{
 @impactTolerance = 12000
 @stressTolerance = 10000
}
```

ModuleWheelBogey

Purpose:
Controls bogey-style pivoting wheel assemblies for load sharing.

Common Fields:

- bogeyAngleLimit (float)
- bogeyResponse (float)
- useBogey (bool)

Runtime Notes:

- Common on large landing gear
- Too much angle causes jitter

Example ModuleManager CFG:
```
@MODULE[ModuleWheelBogey]
{
 @bogeyAngleLimit = 15
 @bogeyResponse = 4
 @useBogey = true
}
```

ModuleWheelLock

Purpose:
Locks wheel movement to stabilize vessels on the ground.

Common Fields:

- lockEnabled (bool)
- lockTorque (float)
- autoLock (bool)

Runtime Notes:

- Do not engage at speed

Example ModuleManager CFG:
```
@MODULE[ModuleWheelLock]
{
 @lockEnabled = true
 @lockTorque = 200
 @autoLock = false
}
```

Appendix A — Common Wheel Tuning Recipes

This appendix provides practical, field-tested tuning recipes. Values are starting points; always validate in-game under expected mass, gravity, and speed conditions.

Light Rover (Mun / Minmus)

Optimized for low gravity, low speeds, and stability on uneven terrain.

Recommended Settings:

- Suspension: moderate travel, soft springs
- Steering: aggressive at low speed, limited at high speed
- Motor torque: low-to-moderate
- Bogey: small angle limit for stability
- Wheel lock: disabled

Example ModuleManager CFG:

```
@MODULE[ModuleWheelSuspension]
{
 @suspensionDistance = 0.25
 @springRatio = 5
 @damperRatio = 1
}

@MODULE[ModuleWheelMotor]
{
 @maxTorque = 12
}
```

Heavy Rover (Duna / Eve)

Designed for higher gravity and mass with emphasis on traction and damping.

Recommended Settings:

- Suspension: longer travel, stiff springs
- Damper ratio increased to prevent oscillation
- Motor torque scaled to mass
- Bogey enabled for load sharing

Example ModuleManager CFG:

```
@MODULE[ModuleWheelSuspension]
{
 @suspensionDistance = 0.4
 @springRatio = 10
```

```
  @damperRatio = 2
}

@MODULE[ModuleWheelBogey]
{
 @bogeyAngleLimit = 18
 @bogeyResponse = 3
}
```

Aircraft Main Landing Gear

Absorbs vertical touchdown loads while maintaining roll stability.

Recommended Settings:

- Suspension: long travel, stiff springs
- High damper ratio
- Steering disabled
- Wheel lock disabled on touchdown

Example ModuleManager CFG:

```
@MODULE[ModuleWheelSuspension]
{
 @suspensionDistance = 0.6
 @springRatio = 15
 @damperRatio = 3
}

@MODULE[ModuleWheelSteering]
{
 @steeringEnabled = false
}
```

Aircraft Nose Gear

Provides steering authority while minimizing shimmy.

Recommended Settings:

- Shorter suspension travel than main gear
- Reduced steering response
- Optional auto-lock after landing

Example ModuleManager CFG:

```
@MODULE[ModuleWheelSuspension]
{
 @suspensionDistance = 0.3
}

@MODULE[ModuleWheelSteering]
{
 @steeringResponse = 3
}
```

Hard-Landing Spaceplane / SSTO

Designed to survive high sink-rate landings.

Recommended Settings:

- Very stiff springs and high damper ratio
- High damage tolerances
- Wheel lock available after rollout

Example ModuleManager CFG:

```
@MODULE[ModuleWheelSuspension]
{
 @springRatio = 20
 @damperRatio = 4
}

@MODULE[ModuleWheelDamage]
{
 @impactTolerance = 15000
}
```

Appendix B — Troubleshooting Decision Table

Use this table to quickly identify which parameters to adjust when wheels behave incorrectly.

Observed Symptom	Likely Cause	Recommended Adjustments
Vehicle bounces repeatedly	Spring too stiff or damper too low	Lower springRatio; increase damperRatio
Vehicle bottoms out	Suspension travel too short or springs too soft	Increase suspensionDistance; raise springRatio
Rover flips during turns	Too much steering at speed; high center of mass	Reduce steeringCurve at speed; lower antiRoll
Wheel shimmy / oscillation	Damper too low; bogey too responsive	Increase damperRatio; reduce bogeyResponse
Landing gear collapses on touchdown	Impact tolerance too low; springs too soft	Increase impactTolerance; raise springRatio
Craft slides while braking	Brake torque too high for friction	Lower maxBrakeTorque; increase frictionMultiplier
Gear locks cause instability	Lock engaged at speed	Disable autoLock; engage lock only when stopped

Appendix C — Presets

These presets group commonly adjusted values by craft type. They are intended as quick baselines.

Rover Preset

1. @MODULE[ModuleWheelSuspension] { @springRatio = 6 @damperRatio = 1.5 }
2. @MODULE[ModuleWheelMotor] { @maxTorque = 15 }
3. @MODULE[ModuleWheelSteering] { @steeringResponse = 5 }

Aircraft Main Gear Preset

4. @MODULE[ModuleWheelSuspension] { @springRatio = 14 @damperRatio = 3 }
5. @MODULE[ModuleWheelSteering] { @steeringEnabled = false }

Aircraft Nose Gear Preset

6. @MODULE[ModuleWheelSuspension] { @springRatio = 10 }
7. @MODULE[ModuleWheelSteering] { @steeringResponse = 3 }

Heavy Landing Preset

8. @MODULE[ModuleWheelSuspension] { @springRatio = 20 @damperRatio = 4 }
9. @MODULE[ModuleWheelDamage] { @impactTolerance = 15000 }

Appendix D — WheelTunerWindow Mapping

Goal	WheelTunerWindow Control
Control bounce	ModuleWheelSuspension → damperRatio
Increase ride height	ModuleWheelSuspension → suspensionDistance
Limit steering at speed	ModuleWheelSteering → steeringCurve
Increase traction	ModuleWheelBase → frictionMultiplier
Ground anchoring	ModuleWheelLock → lockEnabled