

[1.7.x] KSPWheel - Physics Based Alternate Wheel Collider [API Only]

By Shadowmage, November 30, 2016 in Add-on Releases

• api

• wheels

Reply to this topic

1

NEXT

»

Page 1 of 21

Posted November 30, 2016



KSPWheel - An alternate physics-based wheel collider solution for Kerbal Space Program.

This is not a parts mod. It does not add new parts to the game. It is useless by itself.

What it **is**, is a new Wheel Collider component, and a set of PartModules to enable its use on stock-rigging-compatible wheel models. It is intended to be packaged as a dependency with other mods, similar to how FireSpitter / ModuleManager are used.

What it **does** is provide a new physics-based wheel collider that is not subject to the limitations or performance constraints of the Unity5 WheelCollider component or the stock KSP PartModules. It allows for multiple colliders in a single model with arbitrary collider orientations. It uses a discrete spring/damper based suspension model that properly responds to changes in wheel loading (add more weight, and the suspension compresses more) (also now includes an optional auto-spring-calculation mode). Includes curve-based independently tune-able lateral and longitudinal friction calculations, configurable rolling resistance, accurate and configurable motor power use calculations, and configurable steering parameters.

There is nothing here intended for end-user use. This plugin is an API and consists purely of functions and features intended for other mods and modders to use.

This thread is intended as a centralized area for discussion of features and balance. DO

NOT use it to submit bug reports. You may -check- with others to see if something is a bug through the forum thread, but if it turns out to be a bug, you -must- file a report through github; failure to properly file the bug report will result in the bug not being fixed.

Downloads: <https://github.com/shadowmage45/KSPWheel/releases>

Documentation: <https://github.com/shadowmage45/KSPWheel/wiki>

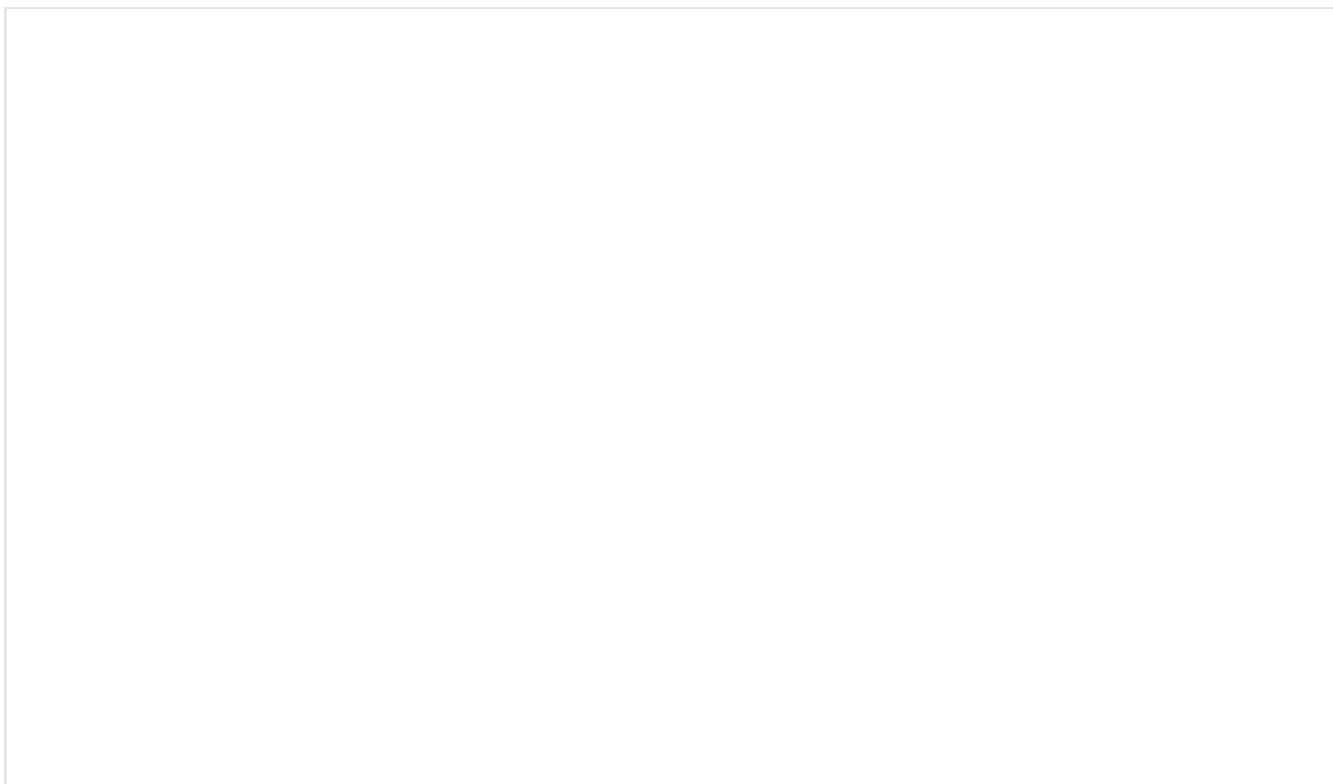
Bug Reports: [Only accepted through Github](#). Include logs (KSP.log), screenshots if applicable, and steps for duplication or your issue will be summarily closed without response. Be polite, do the responsible thing, and file proper bug reports. Bugs reported on the forums will result in being reported to moderators for cluttering the thread and being personally ignored. Just don't do it. If you can't be bothered to submit a proper bug report, I'm not going to bother to fix your problem.

Source: <https://github.com/shadowmage45/KSPWheel>

License: Source is licensed under GPL. Please see the accompanying license files, browse them at the repository, or view them on the web at: <https://www.gnu.org/licenses/gpl-3.0.en.html>

Legal / Other: This mod redistributes ModuleManager under its own license terms.

KSPWheel is Used in the following mods (pm me to be added to the list):



Edited April 14 by Shadowmage

 [Quote](#)

Sharpy, blackheart612, Stone Blue and 51 others like this



Posted November 30, 2016



Post reserved for future additions (docs / whatever)

 [Quote](#)



Posted November 30, 2016



Thank you very, very much. I'm looking forward to seeing how SSTU and Kerbal Foundries work with this collider!


 [Quote](#)



Posted November 30, 2016



May the Kraken bless you.

 [Quote](#)



Posted November 30, 2016



Hi, congrats on the very welcome beta. In the OP you refer to non standard wheel arrangements, I have a few items that in previous lives had two independent wheel units, could you perhaps give an example of how you would set something like that up in the cfg, i took a look at the cfg for the KF mini track, that being the closest similar thing, and see that all that seems to be required is that the WC's have unique names, and the code does the rest, could you confirm ? Cheers



Quote

Shadowmage likes this



Posted November 30, 2016



▼ On 11/30/2016 at 6:24 PM, SpannerMonkey(smce) said:



Hi, congrats on the very welcome beta. In the OP you refer to non standard wheel arrangements, I have a few items that in previous lives had two independent wheel units, could you perhaps give an example of how you would set something like that up in the cfg, i took a look at the cfg for the KF mini track, that being the closest similar thing, and see that all that seems to be required is that the WC's have unique names, and the code does the rest, could you confirm ? Cheers

That is indeed the case. Merely make sure your wheel-collider (and suspension, wheel-mesh, etc) transforms have unique names, and the code will handle the rest. In the future it will even support multiple same-named transforms within a part/model (for use with part-welding; will use an 'indexInDuplicates' to determine which of the same-named transforms to use), however names must currently be unique.

Transform orientations are mostly the same as stock, and most are also configurable (e.g. the suspension mesh movement axis, wheel rotation axis, wheel-bogey/foot axis). So far it has been fully compatible with all stock wheel/leg/gear models, but many have required specialized configs to deal with the inconsistencies in the stock model rigging (some have wheel colliders at the top of suspension, some in the middle, some at the bottom; some have suspension start compressed, others extended).



Quote

SpannerMonkey(smce), Stone Blue and colmo like this



Posted November 30, 2016



Hoping to have time to try this out. Just gotta finalize a very non-wheel related mod. Off road vehicles that can traverse non-flat landscapes would be most welcome. :-)

 [Quote](#)

Fireheart318 likes this



Posted November 30, 2016



Will this allow a rocker-bogie suspension in a single assembly (ship)?


I currently am experimenting with such a suspension, but it requires 5 separate assemblies connected by colliders just for the carriage. The rover itself is a sixth assembly.

 [Quote](#)



Posted December 1, 2016



 **On 11/30/2016 at 11:52 PM, ioreult said:**



Will this allow a rocker-bogie suspension in a single assembly (ship)?

I currently am experimenting with such a suspension, but it requires 5 separate assemblies connected by colliders just for the carriage. The rover itself is a sixth assembly.

It doesn't explicitly disallow it in a single part, but it may require some additional coding support to manage rotating the suspension properly to average the wheel collider compression lengths, and no guarantee that the colliders will play nicely with being manually rotated/moved. I'll give it some thought; might be doable, but will be hard to test without any pre-rigged models.

Did some tweaks to the friction model today, namely the method of combining the sideways and forward friction to ensure it doesn't exceed what the tire can produce. The previous method was set to give precedence to forward friction, and used the post-slip maximums as caps; the result was that cornering force was greatly reduced at high forward velocities, and it was very easy to cause the drive wheels to break loose as soon as any sideways force was applied. It also limited the maximum sideways friction to roughly

half of what it should have been under many circumstances.

The new model uses the maximum peak friction of the tire as determined by the peaks in its friction curves, and limits both forward and sideways frictions proportionately. This results in far greater cornering force at higher velocities, overall increased sideways friction, and a decreased tendency for drive-wheel slip. Overall handling response is greatly increased, though the increased sideways friction may cause a greater tendency for rolling and oversteer at higher velocities with high slip angles or steering angles... sort of like real life.

Another small tweak to the physics code was to re-enable the acceleration from gravity along the forward direction as long as the brakes are off. This is part of the sticky friction implementation and is responsible for keeping the vehicle from sliding on hills; only when it is in the forward direction it is rolling, and not sliding, and its good for things to roll. So roll on... just be careful on those steep hills 🤖 (brakes are still OP and should stop all acceleration as soon as used; working on a solution for that, so that the static friction code only kicks in at low velocities)

Also added basic support for wheel sounds. Motor sounds, running sounds, and slip sounds. It works... not much more to say on the functioning. Currently using some KF sounds for testing, and probably won't be including any sounds in the distributions... but at least the support will be there, and stock sounds can still be used in the included configs. Will also be looking into adding deploy/retract sounds to some of the landing gear and landing legs (that was a stock thing added recently, right?).



Quote

Madrias and Stone Blue like this



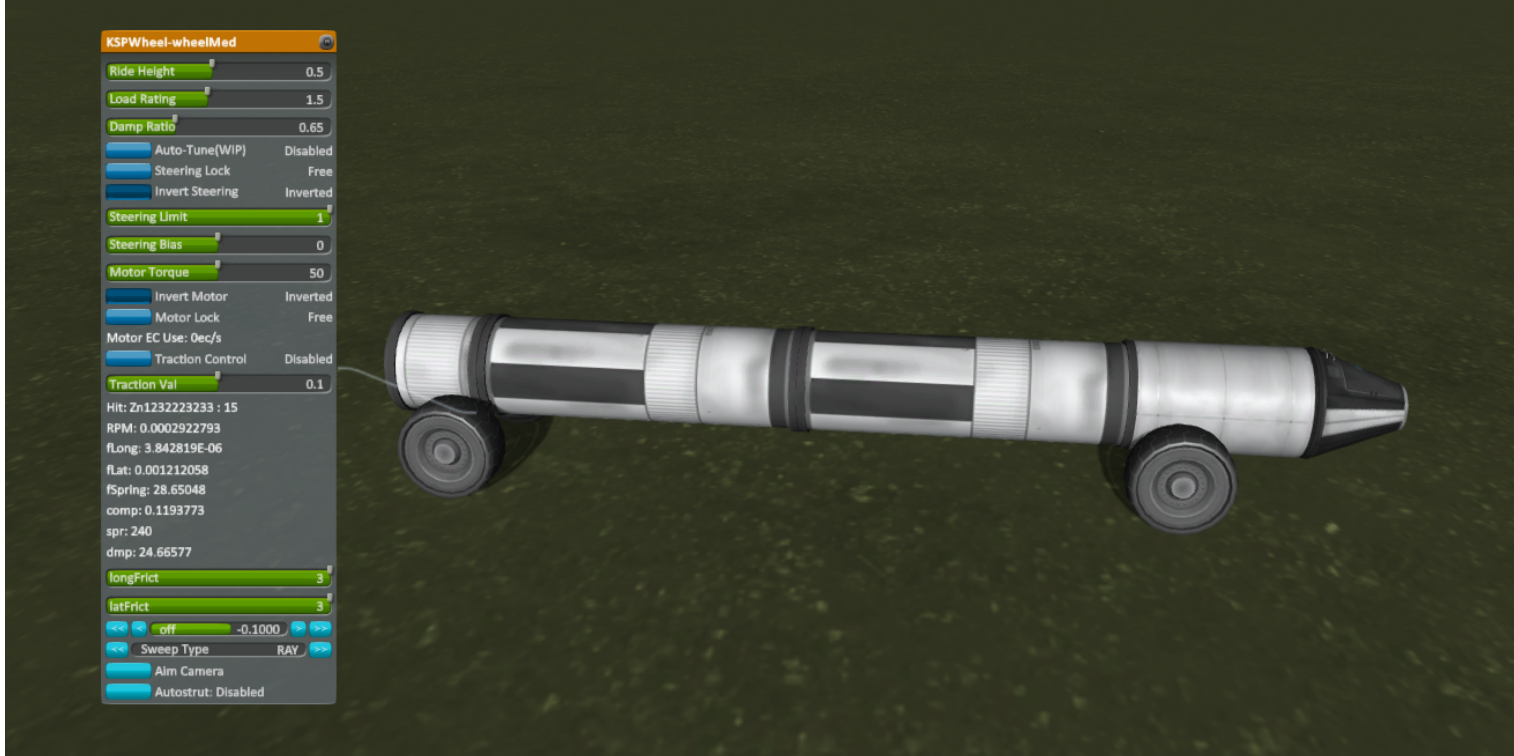
Posted December 1, 2016



great stuff man 🤖

Did some quick testing and here are some thoughts: (edit: tested version = 0.9.0.2)

test vehicle setup; I tried to mimic stock pre 1.1 wheel behaviour that had much higher friction values so most tests are done with max friction values (added benefit is that I can observe wheel reactions to different settings / driving conditions more accurately)



1) would be nice to have higher limit of long/lat friction than 3, I have a feeling that I cannot get to the point of wheels friction of pre 1.1. So even with value of 3 wheels are prone to skidding especially at low speed. from my experience high friction limit is good for low g bodies so you can control your vessel more effectively especially when climbing hills. (i don't find 'flippability' a problem since imho I expect from real world physics and also when steering carefully or driving straight you're fine).

2) pre 1.1 unity had something like wheel weight value which was great for me since it prevented wheel spinning naturally (i used to bump it to high value which worked automatically like traction control), but I guess new physx got rid of this value and I have to tweak traction control values now for no-spin wheels

3) I can't get into the sweet spot of high damp ratio and high ride height (or high loading ratio) since all those values seems to increase fSpring values and when fSpring gets above 30 (in this case) it starts to freak out (weird range of values) and it ends up in a bouncy wheel effect (bouncing vehicle into air etc). Personally I prefer big damping value and medium to high ride height / loading ratio since it evens out high speed driving on uneven terrain and recovering from landing after bigger or lower jumps. Big loading ratio is also nice for heavier industrial rovers so the springs aren't compressed to the max and can do some hops damping work.

4) another nice to have would be to have selectable wheel groups - something like in pre 1.1 KF code, so you could tweak values for all wheels in the same group in editor or in flight. It would make testing much faster especially for vessels with more wheels

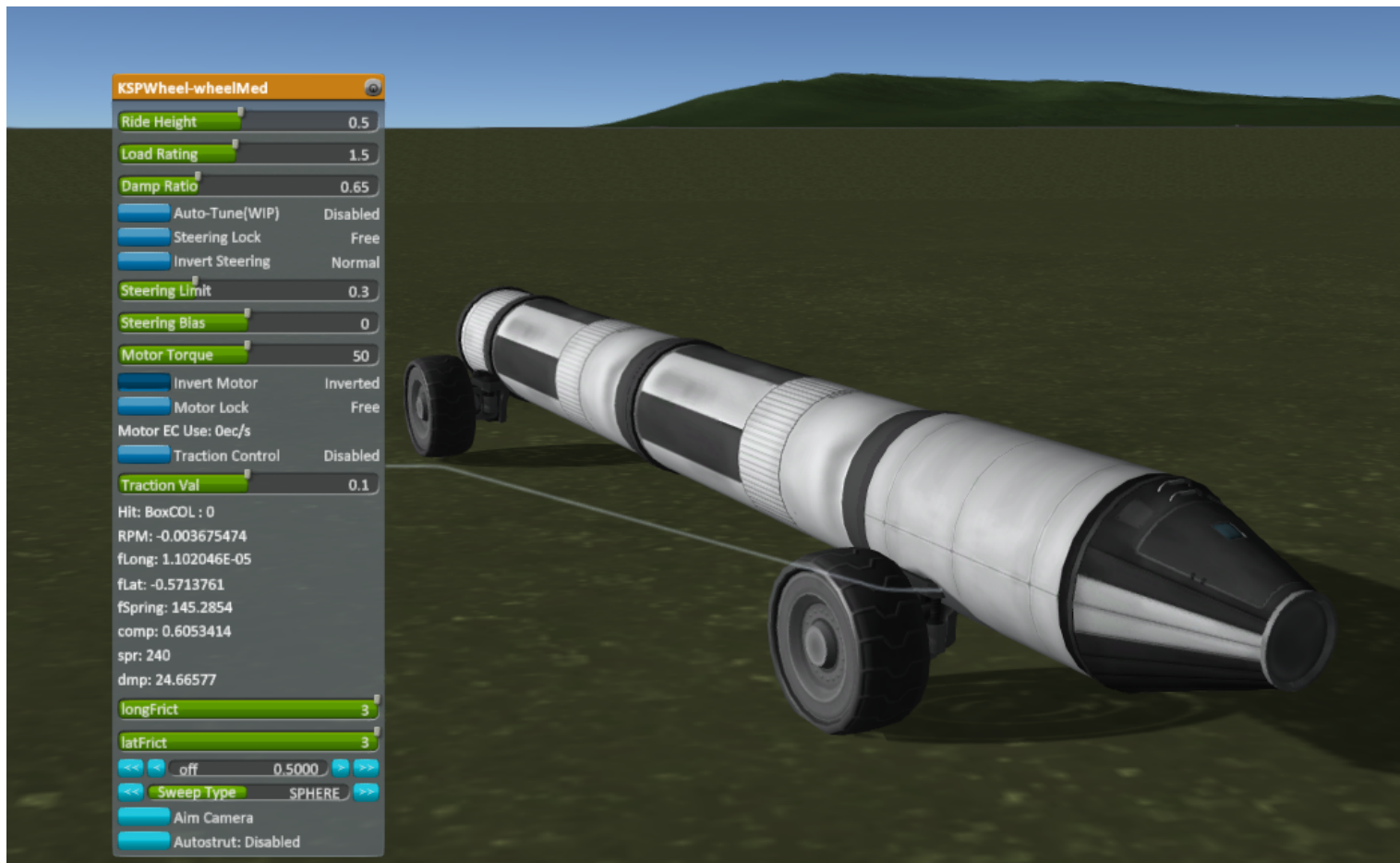
5) haven't tested if but did disabling wheel collider disables steering as well (i.e. when wheels are stowed), would be good to have for possible deployable wheels so they don't turn when they are not deployed

6) tr-2l wheels that i tested seems to be excessively sensitive to angle change on terrain. I.e. when driving out or onto runway (small slope) even with slow speed they bounce of terrain or get stuck and explode. changing wheel collider to capsule or sphere doesn't

change anything. You can replicate that just making circles on the runway driving into runway side slopes.

7) I feel it would be good to have a steering limit written as a curve that depends on speed somewhere in the future 🤖

8) sometimes changing colliders from ray to sphere causes the wheel to go up (to look right again I had to use offset 0.5) and then it stops behaving normally (can't turn properly and then becomes bouncy and itchy) example setup when this happens to front wheels:



9a) something is weird with flipping the vehicle with ray or capsule colliders. I simply cannot flip the vehicle naturally turning hard with max friction. What I mean is that the vehicle comes to about 30 degree flip and then the flip just stops there unnaturally. The effect is similar to games when your roll is artificially reduced so you can't flip it (I think WoT used to have this before physics overhaul). vehicle Pic:



9b) the same vessel that is much lighter (without fuel in tanks) behaves weirdly as well - rolls to about 45 and then is blocked by some imaginary roll force then. It can be overcome by turning even harder so there's something like natural flip to 45 degrees - block at that range - flipping further if more roll force is given. Maybe it's because the wheels apply force to the body? I don't know.

10) decreasing loading rating to a small value does something weird to steering of front wheels - they understeer heavily no matter what speed or whether they exceed friction level or not.

edit2: tried similar tests with most recent dev version and results are similar.

So yah again great work fighting that long with the broken physX of unity 5, one has to be amazed (not) how much they fubared wheels physics that worked great in unity 4 :/

Edited December 1, 2016 by riocrokite



Quote

Stone Blue likes this



Posted December 1, 2016



▼ On 12/1/2016 at 9:45 AM, riocrokite said:



[...]

Did some quick testing and here are some thoughts: (edit: tested version = 0.9.0.2)

[...]

So yah again great work fighting that long with the broken physX of unity 5, one has to be amazed (not) how much they fubared wheels physics that worked great in unity 4 :/

Thanks for the feedback. Much of that is either known issues, recently fixed, or nothing I can do about.

To start, you should read the post I made last night about tweaking friction; it solves many of the problems you had.

1.) Those are debug controls and will be going away entirely in the near future. Don't rely on them, they were intended to help me tweak friction curves and give data for testing. Overall sideways friction is greatly increased in the dev builds from last night. Generally, I thought the U4 wheels had **terrible** sideways friction and rolled vehicles far too easily, so I hope that my wheels aren't anywhere close to that; however what you are seeing in the 0.9.0.2 build is... not correct either.

2.) WheelMass still exists, but is a config only setting. Technically I could open it up to user config, and might. It does pretty much what you would think; makes the wheel more resistant to torques, so each unit of torque applies less rotational speed (though the same net force), and results in better wheel response to motor input, less easy to break the drive wheels loose.

3.) Your not going to. Due to limitations in integration, high damping rates or high ride heights will cause problems with the physics. There is literally zero that I can do about this as I don't have access to the unity physics integration sub-steps; I only have access to FixedUpdate. Trying to get the vehicle to ride at the top of its suspension will never work right; it drives the spring forces too high and will cause bouncing. Raising the damper ratio too high will result in jitter due to the low simulation frequency. The wheel physics want 1000+hz, but KSP/Unity only gives them 50hz.

4.) Thinking on it a bit, and might do something like that; but these modules are not intended to replace KF, which will still have all of its old features/functions.

5.) Noted, yea, they shouldn't steer in the stowed state (stowing does disable the wheel collider / physics updates).

6.) Sounds like a spring/ride height configuration problem. Would have to see the craft and wheel setup. Setting up the load rating, ride height, and damper value properly is **everything** for these wheels. Failure to do so will result in bad results. Sadly I cannot do it automatically as the proper values are different on a per-craft/vehicle basis, nor do I believe it is proper for a wheel to magically change properties.

7.) Its already in there, and enabled by default for all wheels. However it needs to be customized on a per-wheel basis, which I have not the time (or need, or desire) to

implement. The PR button is that way ==>

<https://github.com/shadowmage45/KSPWheel/compare?expand=1> (curve starts at 10m/s, 100% output, down to 50% output at 30m/s). (I use a flight stick for input, so fine analog input is not a problem for me; steering curves actually get in my way as I expect linear steering response regardless of speed).

8.) Sounds like you had the sphere colliders/spherecast hitting the vehicle somewhere else/another part. They are spheres with radius of the wheel, so if they would contact the other bits of the vehicle at all in their suspension sweep, they'll exert force to the vehicle itself. TL:DR -- known problem with sphere casts as they are often much wider than the tire; they solve many problems, but bring with them problems of their own. That is why the sweep-type option exists; a single setting is not correct for every wheel or use. Would have to load the vehicle up to examine clearance.

9.) Sounds like a combination of the friction problem I cleaned up last night, along with one of the suspension quirks that I haven't yet removed (it decreases suspension output with the dot of the angle between the hit normal and suspension angle). Will be removing that quirk as I have found that yes, it interferes with proper force output when 'about to roll'.

10.) Sounds like you decreased the suspension too far and hit the bump stop collider. If your wheels are at max compression / hitting bump stop, you will get incorrect force output as it is impossible to calculate the spring force unless the body is fully supported by the spring; if you are on the BS collider, you're not on the spring. Intentional effect -- don't overload your wheels, and your friction will be correct 🤖 (also part of this is that same friction tweak I mentioned last night). Also, as you accelerate, the mass distribution shifts rearward, reducing available friction from the front tires.

Note, if you want to try dev versions, you need to grab from the dev4 branch (plugin here: <https://github.com/shadowmage45/KSPWheel/blob/dev4/GameData/KSPWheel/Plugin/KSPWheel.dll>). I would suggest dropping that plugin into your install and giving some of the friction tests another try; you'll probably be pleasantly surprised.

Some of the physics problems I can do nothing about. Wish I could, but sadly, nothing I can do given the limited interface I have with the Unity/PhysX code and the super-low simulation frequency in use. Seriously, try the code in the Unity editor at 1000hz and you can just watch 90% of these problems disappear (no more jitter at high damp ratios, no more bounce from high spring rates or ride heights).

Edited December 1, 2016 by Shadowmage



Quote

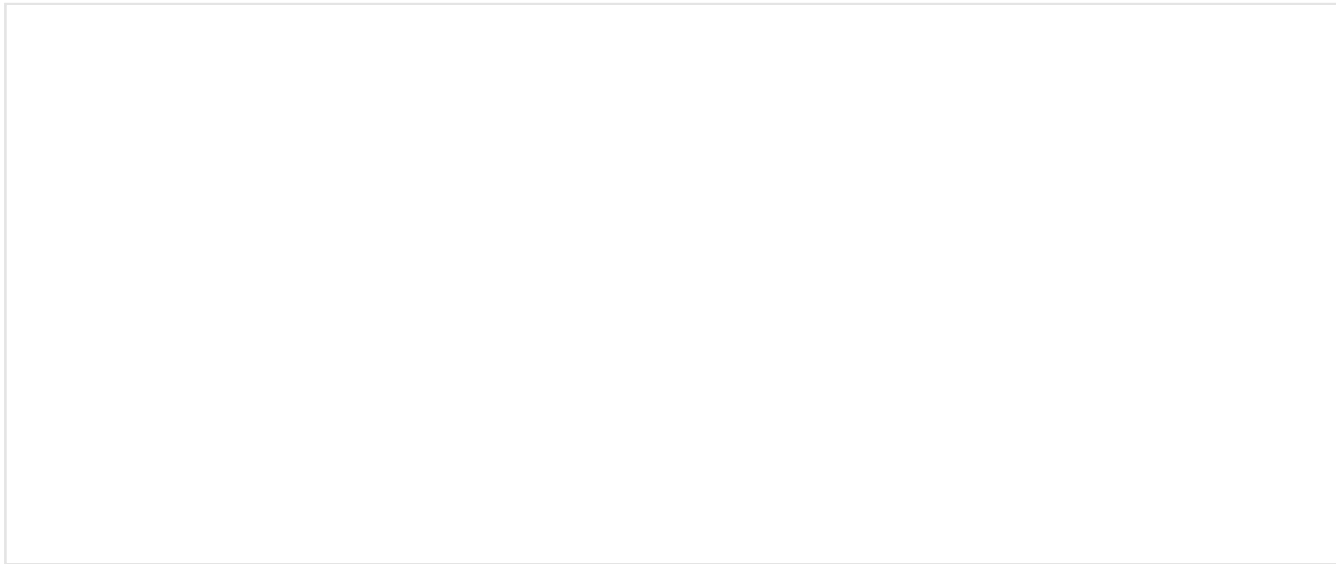
Stone Blue, riocrokite and Beetlecat like this



Posted December 1, 2016



Woooo!!! Thank you for your amazing contributions Shadowmage!! hopefully we will see a return of Adjustable Landing Gear, although according to [@BahamutoD](#)'s profile they haven't been on since July, if Baha doesn't return here is to hoping one of the other talented members of our community picks this up, if only I had the time to teach my self C# 🤖



Quote

DirkLarien likes this



Posted December 1, 2016



▼ On 12/1/2016 at 3:39 PM, Akira_R said:



Woooo!!! Thank you for your amazing contributions Shadowmage!! hopefully we will see a return of Adjustable Landing Gear, although according to [@BahamutoD](#)'s profile they haven't been on since July, if Baha doesn't return here is to hoping one of the other talented members of our community picks this up, if only I had the time to teach my self C# 🤖

I'd have to take a look at the models and plugins for that mod; but things -should- be compatible, at least at the collider level (may require adjustments to other modules / custom plugins to manage the rest of the functions).

Thankfully he had licensed it as CC-XXXX, so some continuation/reboot may be possible even if BD doesn't return.

From a brief look at the BD code that was in use... yes, it could probably use the existing models, but it would need entirely new driving plugin code to implement some of the features (the existing BD animation code leans heavily on the U4 wheel collider and all of

its quirks). Seems like some of the major portions of that mod are already included in KSPWheels though -- ride height, spring, damper.

So yes, should be compatible, but it wouldn't be using any of the BD modules/plugin, at least not if I were to do it (would end up being a simple patch set, as I've done for the KF parts). Scaling would be implemented through tweakscale.

Of course if BD ever came back, he would be more than welcome to integrate the new collider into his existing code (and it should be doable without too much trouble), but updating others' mods is far outside the scope of what I'm trying to accomplish here.

Edit:

Will probably have an updated release this evening with a few more fixes (notably the friction fix, and proper part.rescaleFactor handling), feature enhancements (sounds!), and many updated configs (load ratings, ranges, motor torques, wheel masses).

Closing in pretty fast on finishing off the currently planned TODO and to-fix lists; a few more module features to implement, and a few more physics-related problems to try and sort through, but the majority of features and functions are in place and working as intended (or as good as it can given the integration limitations). Hoping to have things mostly wrapped up in the next week or so, and from there on development will move to a 'maintenance mode' focusing on bug-fixing and stability.

One thing that others can do to help the process would be to outline what you think are reasonable load, max rpm, and torque limits for all of the various parts. Some of them I'm quite stumped as to what their intended range of uses is, such as the airplane landing gear (how much mass should the largest gear support, maxed out? How about the large, medium, and the rest?).

Edited December 1, 2016 by Shadowmage



Quote

Akira_R likes this



Posted December 2, 2016



@Shadowmage if you could post a "lemp fix" to adjustable landing gear using your awesome shadow magic you would be a hero, many people in the [RO/RSS](#) community relied on these to build with.

And perhaps your mod could become the new standard for [RO/RSS](#) wheels if your willing to have them become so. ...drifting planes down runways due to wheel "bug" issues is heartbreaking when your trying to avoid "reverts" in a hard core RP-0 playthrough

 Quote

Feni and crapstar like this



Posted December 2, 2016



Nothing to really add as I haven't tried this well needed Mod. Why this has to come from the community, from people that have RL jobs exc, is a complete mystery to me. I'm absolutely ticked off at the lack of professionalism from companies producing games these days. Thank you very much for this attempt and I'm sure completion of this Mod at some point. I'm a huge fan of exploration with rovers on all the body's we have in KSP and Mods that create more fun . I tweaked some of Rover's wheels, as well as the stock wheels to be much more compliant like a "Rock Crawler", but you can only get so far until it starts becoming unmanageable. Looking forward to using Kerbal Foundries pack with this. Now to end this post, people with real lives, busy lives, turn out better work than the paid teams that claim to be professionals. I'm a bit older than 95% of this community so I have a bit of a different perspective on this. Not to say I know more, or am somehow wiser, its just that I have watched the decline over the last 20 years and its just a crock of Bull Droppings, or Human waist, or maybe greedy a\$\$ hats. (Sorry, the language sensors drive me nuts)So looking forward to proper wheel and track behavior.

Thx man

Edited December 2, 2016 by ArkaelDren

 Quote

Shadowmage, Akira_R and riocrokite like this




Posted December 2, 2016



Wonderful development!

The way I see it, after reading the topic and watching the screenshots, it looks to me there's an important feature missing which will greatly enhance vehicle behaviour and will solve most problems if possible: adjustable bump and rebound of the damper. I've also asked for this in the stock game.

Furthermore I'd like to show something which might help with the ride height/spring behaviour, if only for showing what a magnificent system it is (I have it on my car). Gas springs work differently than metal springs but I have yet to find the documentation how exactly and I don't know if this is something that can be translated into code.

 Quote

A nitrogen reservoir with variable volume yields a spring with non-linear force-deflection characteristics.

https://en.wikipedia.org/wiki/Hydropneumatic_suspension

Edited December 2, 2016 by Azimech

 Quote

Shadowmage likes this



Posted December 2, 2016



▼ On 12/2/2016 at 4:47 AM, Aazard said:



@Shadowmage if you could post a "lemp fix" to adjustable landing gear using your awesome shadow magic you would be a hero, many people in the RO/RSS community relied on these to build with.

And perhaps your mod could become the new standard for RO/RSS wheels if your willing to have them become so. ...drifting planes down runways due to wheel "bug" issues is heartbreaking when your trying to avoid "reverts" in a hard core RP-0 playthrough

I'll see what I can do about creating some patches to at least enable basic functionality for those parts. Have a fairly full plate at the moment though, so might be awhile before I can get to it.

▼ On 12/2/2016 at 7:10 AM, ArkaelDren said:



Nothing to really add as I haven't tried this well needed Mod. Why this has to come from the community, from people that have RL jobs exc, is a complete mystery to me. I'm absolutely ticked off at the lack of professionalism from companies producing games these days. Thank you very much for this attempt and I'm sure completion of this Mod at some point. I'm a huge fan of exploration with rovers on all the body's we have in KSP and Mods that create more fun . I tweaked some of Rover's wheels, as well as the stock wheels to be much more compliant like a "Rock Crawler", but you can only get so far until it starts becoming unmanageable. Looking forward to using Kerbal Foundries pack with this. Now to end this post, people with real lives,

busy lives, turn out better work than the paid teams that claim to be professionals.

I'm a bit older than 95% of this community so I have a bit of a different perspective on this. Not to say I know more, or am somehow wiser, its just that I have watched the decline over the last 20 years and its just a crock of Bull Droppings, or Human waist, or maybe greedy a\$\$ hats. (Sorry, the language sensors drive me nuts)So looking forward to proper wheel and track behavior.

Thx man

Thanks for the support 🤖

I wouldn't hold things against SQUAD so much; they did the best that they could given the pile of garbage that Unity handed them. But yeah, kind of sad that solutions to game-engine-level problems have to come from a modding community for a single game using that engine; the problems really need to be solved at the engine level (read: Unity should have never released such a limited system).

▼ On 12/2/2016 at 7:13 AM, Azimech said:



Wonderful development!

The way I see it, after reading the topic and watching the screenshots, it looks to me there's an important feature missing which will greatly enhance vehicle behaviour and will solve most problems if possible: adjustable bump and rebound of the damper. I've also asked for this in the stock game.

Furthermore I'd like to show something which might help with the ride height/spring behaviour, if only for showing what a magnificent system it is (I have it on my car).

Gas springs work differently than metal springs but I have yet to find the documentation how exactly and I don't know if this is something that can be translated into code.

https://en.wikipedia.org/wiki/Hydropneumatic_suspension

Interesting find on the gas suspension. Non-linear spring/dampers are certainly a possibility, or even other non-standard suspension calculation methods. Would be quite easy to put in some curves that can be polled to determine spring/damper output for compression input -- would merely need to figure out how to best define those curves from the available real-world data for the different spring types. Will give some thought as to how to best integrate this functionality in a clean manner; can work on figuring out the

curves for the suspension types after the feature is working 🤖

Not quite sure what you are referring to with adjustable bump? (As in bump-stop? Not much that can be done to tune it; it uses a collider to ensure overcompression does not occur)

Guessing that damper rebound is referring to using a different damper rate for compression vs. decompression... which I've been thinking myself is a bit of a needed thing. Could benefit the handling for many off road type setups, especially at higher speeds when traversing smaller rough details.

(Disclaimer: I am not a vehicle physics expert, merely a programmer with a decent grasp of maths; many of the specific terms are unfamiliar to me by name, but I can probably tell you how they function...)

 [Quote](#)

Akira_R, riocrokite and Feni like this



Posted December 2, 2016



Could one of you gentlemen instruct me on how to create the tracks on the wheels, or is that not implemented yet?

Or do i need KF as a separate mod for that to happen?


Edited December 2, 2016 by mechanicH

 [Quote](#)



Posted December 2, 2016



 **On 12/2/2016 at 3:55 PM, mechanicH said:**



Could one of you gentlemen instruct me on how to create the tracks on the wheels, or is that not implemented yet?

Or do i need KF as a separate mod for that to happen?


I'm afraid I'm not quite sure what you are asking?

Are you wanting to know how to rig your own custom models, and how to make tracks

yourself? (Would have to ask Lo-fi or someone who has actually done it)

Or are you wanting to know how to use KSPWheel alongside KerbalFoundries to enable you to test the tracks in game as shown in some of the screenshots? (Simply install the most recent KSPWheel and the 1.05 version of KerbalFoudries; duplicate parts with the name KSPWheel-XXX will be created; use those duplicate parts for testing).

NOTE -- patched KerbalFoundries model support is for **TESTING OF PHYSICS ONLY**. Those parts/configs **will** be removed as soon as the KerbalFoundries plugin/PartModules are working again; so don't get attached to them, and certainly don't use them on any craft that you want to keep around longer than a day or two.

 [Quote](#)

Fireheart318 and Akira_R like this



Posted December 2, 2016



 On 12/2/2016 at 4:15 PM, Shadowmage said:



Or are you wanting to know how to use KSPWheel alongside KerbalFoundries to enable you to test the tracks in game as shown in some of the screenshots

Thanks Mage this is what i was looking for, i wanted to test the tracks out. I saw on KF version 1.9


Edited December 2, 2016 by mechanicH

 [Quote](#)



Posted December 2, 2016



 On 12/2/2016 at 4:19 PM, mechanicH said:



Thanks Mage this is what i was looking for, i wanted to test the tracks out. I saw on KF version 1.9

You want whatever the most recent version of KF was that was built for KSP 1.05; I have no idea what the actual version number of the KF release is.

 [Quote](#)

mechanicH likes this



Posted December 2, 2016



@Shadowmage Oh yeah, that was directed at Unity, not so much Squad. Not to say I don't have my own issues with some of Squads decisions and directions. I just have a hard time seeing three guys working with very little time and resources, coming out with these beautiful, sometimes complicated Mods. While a large corp, with huge resources has a hard time releasing 2-3 parts. (not Squad, although guilty at times of this very thing).



Quote

Shadowmage and Akira_R like this



Posted December 2, 2016



@Shadowmage to my surprise the tracks actually perform very nicely. Im looking forward to the final product. Thank you Mage, your the best.

Edited December 2, 2016 by mechanicH



Quote

Shadowmage likes this



Posted December 2, 2016



▼ On 12/1/2016 at 5:01 PM, Shadowmage said:



One thing that others can do to help the process would be to outline what you think are reasonable load, max rpm, and torque limits for all of the various parts. Some of them I'm quite stumped as to what their intended range of uses is, such as the airplane landing gear (how much mass should the largest gear support, maxed out? How about the large, medium, and the rest?).

As luck would have it I haven't been around the last few days, so I'm just downloading KSP 1.2.1 now. :-/

There is a worked example at the end of this post. There are a number of fudge factors in the post itself which could be tweaked if it turns out to produce results which are comedically bad.

I don't think max torque is a useful place to start. Max power is, which should be roughly proportional to part mass - changed if we feel less or more of the mass of a part might be the motor (for example, track vs. wheel parts). Electric vehicle motors are approaching 10

kW/kg _but_ that is the motor alone. A Tesla S has a maximum of 568kW power output on circa 460kg of drive bits (motor, inverter, differential, wheels, brakes, suspension, rack and pinion but _not_ the battery since that's a separate part in KSP terms too). Now our parts are engineered for aerospace, but they're also engineered for off-road travel with superbly robust suspensions and parts; we might perhaps get 1 kW/kg on wheels and 0.75 kW/kg on tracks.

(What we decide a kW costs in ElectricCharge is another question, but in terms of determining part performance we need to know kW).

Next is the motor's maximum efficient RPM. Larger motors typically have a lower maximum RPM because the centrifugal force generated increases with the size of the rotating parts. A Tesla motor runs at a maximum 16,000 RPM, but that is high for EVs and that is the absolute maximum. I would suggest that a part the diameter of an ordinary motor car wheel (70cm) should run at 8,000 RPM and then making it scale inversely with part linear dimension - twice the size, max 4,000 RPM, and so forth.

Now we can determine the maximum motor torque. The maximum motor torque is that delivered at 3/4 the maximum efficient RPM and full power. (Where does the figure of 3/4 come from? I have just eyeballed some torque v. speed curves and made it up).

Next is to decide a nominal maximum operating speed. If the part isn't tiny (smaller than the wheels on a real-world 4x4), this should be more based on perceptions of the part's intended usage than anything else - tracks generally slower than wheels, etc. If it is tiny the maximum speed should probably be further reduced. (This is the only really subjective part of this process).

Next is to determine the default gear ratio. We know the nominal maximum operating speed and the diameter of the wheel, so we can tell the wheel RPM at maximum speed. The default gear ratio should be such that at the nominal maximum speed, the motor is running at its maximum efficient RPM.

Now we know the default gear ratio we can divide the maximum motor torque by that to determine torque delivered to the wheel.

The user should be able to alter this default gear ratio in the VAB, down to about half or up to about double the default value. This will either increase the maximum torque but decrease the effective maximum speed, or decrease the maximum torque but make it more viable to exceed the nominal maximum speed.

Motor performance is as follows (based on the actual gear ratio):

If we are below 3/4-maximum-RPM, deliver maximum torque. Calculate power used based on RPM.

If we are above 3/4-maximum RPM but below motor max efficient RPM, deliver full power. Calculate torque based on motor RPM.

If we are above motor maximum efficient RPM, multiply maximum power by $(1 - (2 \times \text{excess RPM} / \text{maximum RPM}))$, minimum 0. At 150% of nominal maximum RPM, the motor can deliver no power.

What about rolling resistance? I wrote about this [before](#). A potted summary is I think the force at the wheel contact point should be 0.08 times the weight on the part (ie, the current force on the suspension, but read on), 0.1 times if the part is tracked.

Add to the current force on the suspension 10N per kg of the part to account for losses internal to the part - even on Gilly, grinding the tracks around takes work.

If the springs are bottomed out, double the rolling resistance force (we can't know exactly how much mass they are supporting, but it's probably too much).

If part speed is below 1 m/s, multiply the rolling resistance force by the speed. (This neatly means 0 force at 0 RPM). This is the only fudge for speed - normally it's not necessary because work done against the force will rise proportionately to speed anyway. This just prevents phantom forces at very low speeds.

If the part is going above the nominal maximum speed, increase rolling resistance by multiplying by $(1 + \text{excess speed} / \text{nominal maximum})$.

Given wheel diameter this can be turned into a torque term to be applied counter to the direction of movement.

Here is a worked example (intermediate values have been rounded). Suppose I have a 200kg part with a wheel diameter of 1m. (Rather like the old KF medium wheel, in fact).

Since this is a wheel not a track and doesn't seem to have any inordinately heavy accoutrements, I will give it 1 kW/kg. Maximum power is 200kW.

It's made for roving not racing so I will assign it a nominal maximum speed of 12 m/s. (This is not so restricting; by gearing it up, the user will be able to go at 24 m/s at full power and get somewhere between there and 36 m/s. IME the old KF parts topped out at about 20 m/s.)

Our wheel is 10/7 as big as the typical motor car wheel so the maximum motor RPM will be 5600.

The maximum motor torque is delivered at or below 4200 RPM, corresponding to 9 m/s with default gearing. The torque at the motor will then be 454 N m. [1]

A 1m wheel at 9 m/s makes $9/\pi$ revolutions per second, 172 RPM. The default gearing is a circa 24:1 reduction. The maximum torque at the wheel axle in the default configuration is 11111 N m. (This seems like a huge number at first, but it's not for a heavy offroad

drivetrain, which is what we've got - and we're turning large wheels reducing force at the contact patch. It's huge because we have a powerful motor geared down to provide high torque at low speeds. Four of these beasts (800kg) would accelerate a 5 tonne vehicle at just under 9 m/s/s from a standing start [2]).

In this default configuration, from 0 to 9 m/s the part delivers 11111 N m. Power consumption rises from 0 to 200kW linearly with speed. Above 9 m/s, power consumption stays at 200kW but torque drops off to $(200\text{kW} / \text{wheel revolution rate})$ [3], to 8333 N m at 12 m/s. Above 12 m/s power will start to also drop off - at 15 m/s power is only 100 kW and torque is now $(100\text{ kW} / ((15/\pi) \text{ revolutions per second}))$, 3333 N m.

If the user had doubled the gear reduction in the VAB, they'd get an impressive 22222 N m of torque... but only up to 4.5 m/s, where torque would start to drop off and power will start to drop off at 6 m/s.

If they'd halved the gear reduction, they get 5556 N m. However, that is delivered up to 13.5 m/s, and power only starts to drop off at 24 m/s. They do however take an increased rolling resistance term from 12 m/s upwards.

What about rolling resistance? Suppose four of these are mounted on a 5 tonne vehicle, but we are on Tylo, where the surface gravity is about 8 N/kg. If the mass of the vehicle is currently evenly distributed, 1250 kg rests on the spring on one part, producing a force of 10kN in the spring.

We add to this the gravity-independent term of 10N per kg of the part itself - 2kN, for a total of 12kN. We multiply by 0.08 to get 960N. At speeds above 1 m/s and below 12 m/s (the nominal maximum speed), this will apply a torque of 1920 N m counter to the direction of movement. This seems to be of about the right order of magnitude. It won't trouble the default gearing producing 11111 N m, but the maximum-speed gear reduction is clearly going to struggle as it approaches 24 m/s (4167 N m torque) with the rolling resistance term doubled for overspeed to 3840 N m.

[1] This is the least obvious derivation. [Here](#) is a perfectly good calculator; or one can type "You have: 200 kW / (4200 revolutions per minute)" "You want: N m" into the UNIX program "units"; or one can know that 1W at 1 revolution per second is $1/(2 * \pi)$ Newton metres of torque so 200kW at 4200 RPM (70 RPsecond) is $200000 / (2 * \pi * (46\ 2/3))$ N m = 454 N m.

[2] An implication is that there may want to be a user facility - not restricted to the VAB - to cut maximum torque at low speeds, so that keyboard users don't constantly do wheelies because they have no choice but to floor the throttle. What I'd suggest is that the user can input a limited torque for use at 1 m/s and below, and torque then rises linearly between 1

m/s and 10 m/s.

[3] At first glimpse this doesn't make sense because torque seems to be independent of gearing. It is - in the range where the motor is operating at full power and performance is limited by that. Gearing changes where that power limited range starts and where the motor power starts to drop off.



Quote

SchwinnTropius, Feni, Shadowmage and 2 others like this



Posted December 2, 2016



Nice writing [@damerell](#) ! Really love the amount of effort you've put into this!



Quote



1

NEXT



Page 1 of 21

Reply to this topic...



GO TO TOPIC LISTING

NEXT UNREAD TOPIC



Language ▼ Theme ▼ Privacy Policy Contact Us

©2018 Take-Two Interactive Software, Inc.

Powered by Invision Community