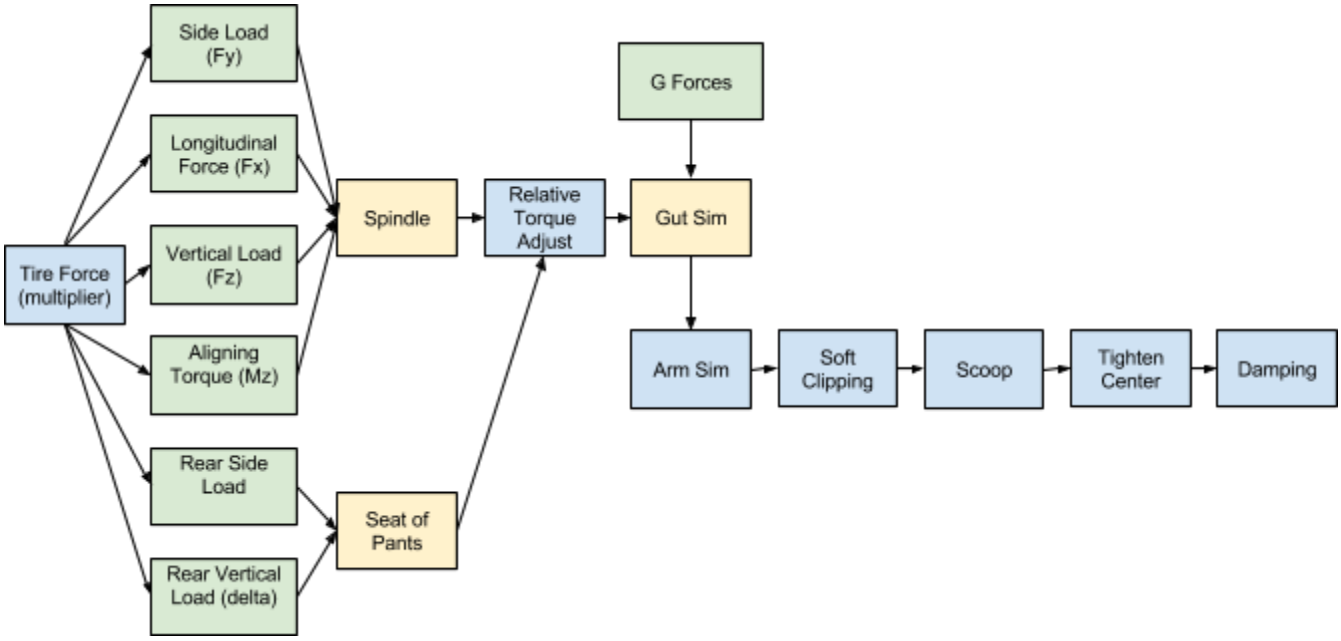
Project CARS Force Feedback Setup Guide

Topology 3

20140725

Design



Key

- green -- input signal
- yellow -- components parameterized as part of car setup
- blue -- components parameterized as part of controller setup

Proposed GUI Exposure

Topology 3 was developed with two goals in mind:

1. Simplify as much as possible such that all parameters maybe be exposed in the GUI
2. Better handle the nuances in the variety of wheels supported

I argue that part of this topology is best served by actually being in the car setups. Two basic goals of setup are balance and feel. FFB is very important to feel, and how a car is setup may benefit from setup specific FFB adjustments. This is fundamentally a concession that simulation racing is a sport similar to, but distinct from, real life racing. In real life racing we deal with things not relevant to sim racing such as seatbelt tension, visor choice, suit ventilation, etc. Or wrenching changes in the hot sun. In sim racing we have FFB latency, dynamic range, and non linearity as some of those things we have to deal with that real life racing does not. So I propose we put the parts of FFB adjustment that couple with traditional setup in with traditional setup logistics. In other words we are saying part of FFB is part of setting up a sim car. This way some FFB parameters will couple, and therefore save and transfer, with the setup they are matched to. I propose this would just be another tab in the setup section.

An example of why this makes sense is the fact that adjusting caster will change FFB magnitude. Caster is inarguably a valid setup parameter. But it is also inarguable that the FFB changes with caster. So it directly follows that a FFB that feels good with a particular caster will need FFB adjustment if the caster is adjusted to feel as good. Other obvious examples include changing tire choice or steering ratio, but almost anything in setup can have similar considerations.

Another case to be made is track considerations. For example, a setup for Montreal might stress Fx more, leveraging limited device dynamic range for heightened braking feel. On the other hand, a setup for Indy might have no Fx at all, to focus in more on on-edge Fy and Mz, or maybe even lean mostly on Seat of Pants and Gut (try it...Indy is pretty immersive and fun that way).

Other parameters are more specific to the controller itself, and these I propose go into the Controller section. The existing Tire Force is already there. Some other scaling and linearity parameters should also go into the Controller section.

Input Signals

The four front tire input signals are the component parts of the whole tire induced torque coming through the rack. So if these are all scaled to the same thing (by convention 1.0), this is the same as straight rack torque.

The two rear tire signals are to enable the Seats of Pants concept. Neither of these go through rack geometry though, as there is no rear rack and steering wheel. These just go straight to the seat.

Finally, the G force signal is to enable the Gut physical simulation concept.

Tire Force

This parameter is already in the Controller section of the GUI, where it should be. This is simply an overall multiplier on all of the input tire forces. Note that G forces, the input to Gut, are not scaled with this parameter. Note that the other FFB parameter in the Controller section is the same as the scaling in the driver. Reducing that does not help saturation, it simply reduces force.

Spindle

Currently these are FFBTweaker parameters, but I propose they become Car Setup parameters.

SpindleMasterScale

This is a multiplier on all of the front tire forces. This was added to allow the following four scale to default to 1, and be more intuitively like “weights”.

SpindleFxScale
SpindleFyScale
SpindleFzScale
SpindleMzScale

Individual scales on the components going through the spindle/rack. To get pure rack forces, leave these all at the same value. 1.0 is a convenient value for this, and use SpindleMasterScale to dial overall spindle force.

SpindleFxLoPass
SpindleFyLoPass
SpindleFzLoPass
SpindleMzLoPass

Individual smoothing on the components going through the spindle/rack. Typically Fx requires more smoothing than the others. 0.0 is no smoothing. 1.0 is normalized to “really smooth but still some useful signal”. Values above 1.0 are valid.

SpindleArm

SpindleArm is the angle, in degrees, of the attachment of the tie rod to the spindle. Zero degrees means the tie rod is attached directly aft of the axis. That particular distance, how far aft, is not critical, because that just amounts to scale, which we adjust based on squeezing into the device range anyway. The angle though matters a lot in how the forces feel when the steering wheel is not straight.

90 degrees is then with the tie rod directly inboard of the axis, which physically would result in the inability to steer. Realistic values I'd guess are between 0 and 45.

Note that the per force Soft Clippers from Topology 2 have been removed. This is one of the adjustments to make FFB control completely from the GUI reasonable.

Seat of Pants

The basic idea of “Seat of Pants” is to present information from what is happening at the rear of the car through force feedback. There are two physical forces that are used. The rear side loads and the rear vertical loads.

Currently these are FFBTweaker parameters, but I propose they become Car Setup parameters.

SoPScale
Overall scaling of Seat of Pants

SoPLateral
Scaling of the rear side load effect.

SoPDifferential
Scaling of the rear vertical load effect, which is actually the difference between right and left vertical loads.

SoPLoPass
Smoothing of the Seat of Pants signal. 0.0 is no smoothing. 1.0 is normalized to “really smooth but still some useful signal”. Values above 1.0 are valid.

Relative Torque Adjust

This is an all new concept for Topology 3. The idea here is to present torque to the wheel based on the change in torque through time instead of as absolute torque. This means that with reasonable parameters, the wheel will never fully saturate. But unlike soft clipping (which can also prevent saturation), the high end torques do not get as heavily squeezed.

There is one side effect to tune out though, and that is the wheel losing center over time. If all torque was completely via “Relative Torque Adjust”, centered torque would move around as the wheel goes through previously saturating torques. To prevent this, we use the bleed value to “bleed” absolute torque back into the mix.

Currently these are FFBTweaker parameters, but I propose they become Controller parameters.

RelativeGain
This is the scaling on the amount of calculated torque change that is applied. 1.0 is the intuitively correct value. 0.0 turns this component off.

RelativeBleed
This is a time value for bleeding absolute torque back in. 1.0s is a good starting point.

RelativeClamp
This is the force to wheel value (so in the 0.0 to 1.0 range) where the non absolute running magnitude is clamped. This does not clamp the overall value, and torques can still go above this, but it does exert a strong clamping effect. 1.0 is a good starting point for this. Values greater than 1.0 can make sense if soft clipping is also used. Values less than 1.0 makes sense to give some headroom for spikes to be a little more symmetrical around the clamp.

Note that with this component on, and with clamp at 1.0 or less, and not too much bleed, there is no full saturation. What this means is that what was too much force before now becomes more force effects felt near full force. But this too can become too much, as that can start to overpower the more subtle unsaturated force range. So you still need to dial overall force (via Tire Force and the scales), but that scaling can become an interesting control, not just something to avoid saturation with.

Gut Simulation

Currently these are FFBTweaker parameters, but I propose they become Car Setup parameters.

This is a simulation of the G forces on the body of the driver. Basically, G forces move the body around via a physical simulation, and the result of that simulation is translated to force feedback.

GutScale
Magnitude of the gut simulation in FFB. 1.0 is normalized to “significant but not overpowering”.

GutLongScale
Magnitude of longitudinal effect applied. This is a scaling of the baseline lateral effect. At 0.0, the gut effect will be all based on lateral G’s. With non zero GutLongScale, under braking G’s, the lateral effect will increase, and under acceleration G’s the lateral effect will decrease.

GutMass
This is the mass of the simulated “gut”, which should not be the whole human body. It should be some lesser portion, roughly being the effective amount of mass not “locked down” rigid by the seat and seatbelts. This is a very fuzzy concept, so the number is really just a very rough

ballpark number. This is fine, because the simulation is not overly sensitive to this number. It matters, but it is not extremely critical.

The default is 50 kg.

Arm Simulation

Currently these are FFBTweaker parameters, but I propose they become Controller parameters.

The arm simulation simulates that the wheel is driven by a non rigid linkage, namely the driver's arms, as well as play and mass in the linkages themselves.. However, this is done purely with force feedback. The position of the the controller still directly dictates the location of the simulation wheel.

This simulation also serves as the main global smoothing stage, which is the main reason this probably belongs in the Controller section and not Car setup.

ArmScale
Ratio of incoming signal to pass through the arm simulation. 0.0 if off. 1.0 is application of all incoming signal.

ArmMass
Mass of "arms", with respect to simulation. This does not necessarily mean the average mass of two human arms. This is the effective mass with respect to the degree of freedom that is the wheel/controller.

ArmStiffness
Spring-like stiffness of the "arms". Stiffer settings will pass through higher frequency information. Softer settings will smooth more.

ArmDamping
This is a multiplier on critical damping of whatever mass and stiffness is set. Therefore, 1.0 means exactly critically damped.

Soft Clipping

Currently these are FFBTweaker parameters, but I propose they become Controller parameters.

This compresses all force within range of the wheel, although the stronger the force, the more it is squeezed into the higher force range. In some ways this is like Log Scaling, but Soft Clipping

guarantees all signal will squeeze into the range, however compressed. On the other hand, approaching linear behavior is not implicit with soft clipping, as it can be with log scaling.

SoftClip
The “half signal” for setting the soft clipper. The value set here is the input signal that will become 0.5 as an output signal. Setting this to 0.0 turns the soft clipping off. Setting this to 0.5 is maybe the closest approximation to linear while on, but is not linear. Setting this to 1.0 will match the derivative/slope of the output at zero input (so if you want the lowest forces to feel similar, and compress everything else). Therefore, less than 1.0 will amplify some lower force, and reduce larger forces. Greater than 1.0 will reduce all forces.

SoftClipUnity
Straight soft clipping will never reach full 1.0 magnitude, which means for lots of soft clipping scenarios, the full force of the wheel is never quite used, possibly to a noticeable level.

SoftClipUnity sets the expected maximum force that will hit the soft clipper, and rescales such that that force outputs at 1.0 (full force of wheel). This means saturation may be reintroduced if this is set too low, but it is useful to fine tune output, especially when the soft clipper is used more for non-linear response than for anti-saturation. Setting this to 0.0 turns the unity re-scaling off.

Scoop

Currently these are FFBTweaker parameters, but I propose they become Controller parameters.

This is a new component for Topology 3, and is directly in response to some devices going flat in response at higher force levels. This is somewhat the opposite non-linear tool as the soft clipper, but is shaped differently, to better fit the nature of devices (and be easier to control).

So what scoop does is reduce lower forces more and high forces less, thereby increasing the slope of force where some devices reduce the slope of force. Since devices seem to do this in two more or less linear regimes, with a knee in between, this is how this component works (in the opposite direction).

ScoopKnee
The input force level where the knee is at. If this is 0.0, this component is turned off.

ScoopReduction
The input force reduction below the knee. Above the knee, the force slope is increased such that at 1.0 input force, the output force is 1.0.

Tighten Center

Currently these are FFBTweaker parameters, but I propose they become Controller parameters.

Note that the name of this can be confusing. This has nothing to do with tightening the wheel about geometric top center. The “center” for this component means “zero force”, and has nothing to do with wheel position.

The primary purpose of this is to remove wheel deadzones, but it cab also be a shaping tool.

TightenCenterRange
This is the input force below which the output force is increased to remove a deadzone. Put more simply, this is the size of the deadzone you are trying to remove.

TightenCenterFalloff
This controls how sharply the output force approaches zero force as the input force goes below TightenCenterRange.

Damping

One use of damping can be to counter inherent drag in a device by using negative BaseDrag. However, often devices do not have linear inherent drag, so setting BaseDrag such that there is little to no device resistance at slow wheel speed will result in accelerating forces at higher wheel speeds. This can be fixed by also having some positive BaseDragSqr.

A technique to set damping to cancel most device drag is to turn off ALL forces (F1 menu, Slow Speed Force, and TireForce) and adjust BaseDrag and BaseDragSqr such that the wheel stays the same speed or slows down ever so slightly (until it hits a stop) when you give it a good push at different rates. It seems better to have a tiny bit of drag left than to have the wheel accelerate on its own at any speed.

BaseDrag
This is resistance on the wheel as a function of wheel angular velocity.

BaseDragSqr
This is resistance on the wheel as a function of wheel angular velocity squared.

BaseDragLoPass
This is smoothing of the angular velocity for drag calculations. Raw position data on some devices can be noisy. Not that increasing smoothing can have a secondary apparent effect of increasing the effect of drag.