Skid-steer friction calibration protocol for digital twin creation

Rachel Trimble[1] and Charles Fox[2]

[1]Department of Engineering, University of Cambridge
[2]School of Computer Science, University of Lincoln

Abstract. Mobile robots require digital twins to test and learn algorithms while minimising the difficulty, expense and risk of physical trials. Most mobile robots use wheels, which are notoriously difficult to simulate accurately due to friction. Physics engines approximate complex tribology using simplified models which can result in unrealistic behaviors such as inability to turn or sliding sideways down small slopes. Methods exist to characterise friction properties of skid steer vehicles [1] but use has been limited because they require expensive measurement equipment or physics models not available in common simulators. We present a new simple protocol to obtain dynamic friction parameters from physical four-wheeled skid-steer robots for use in the Gazebo robot simulator using ODE (Open Dynamics Engine), assuming only that calibrated IMU (Inertial Measurement Unit) and odometry, and vehicle and wheel weights and geometry are available.

1 Applicability of Friction Models in Gazebo

Coulomb friction models the typical observation that dry friction F_C opposes an external pulling force F_e up to a limit proportional (μ) to the normal force F_n at the contact, $F_C = \min(\mu F_n, F_e)$. F_n is the reaction of the supporting surface to the weight of the object. This is often visualised as a cone, such that the object moves (or a wheel loses traction) when the resultant force vector is outside the cone. In ODE this cone may be approximated by a pyramid which is faster to compute. ODE also allows different μ values in longitudinal and latitudinal directions, specified separately as $fdir$. If a vehicle does not need to slip or skid (e.g. it has *two* wheel differential drive and does not travel at high speeds) then a high μ value alone would give reliable movement without slip. The μ values control *maximum* friction forces which can be applied, which are rarely needed in full, so for wheels, where friction is generally desirable, it is common to set them to infinity or 'any high number' > 1. This model is not sufficient in the case of four-wheel skid-steer robots because the wheels do not point in the direction of motion as the robot turns and the wheels simultaneously drive and slip.

Slip friction (FDS) models observed friction in contacts mediated by a lubricating liquid or sand-like particles. Slip friction force is proportional to the *sliding velocity*, $F_S = \frac{v_{slip}}{k}$. ODE allows two separate k slip values for perpendicular directions defined by $fdir$. Using these in longitudinal and lateral directions is a potential mechanism for skid steer vehicles to turn and is a valid approach for deformable surfaces such as sand, but is not realistic from vehicles on hard surfaces because it allows a stationary vehicle to slip sideways down infinitesimal slopes. Most robots are best programmed to avoid lubricated surfaces so this model is not our focus.

wheel_slip is a linearized approximation to the brush tyre model [2], which assumes the wheels are made of stretchable tyres approximated as flexible brushes. In this model, the slip is caused by new tyre elements hitting the road and then deforming as the wheel turns. Hence, a non-turning wheel will not slip and a turning wheel will slip with a slip constant is proportional to the *rotation speed* (v_r) of the wheel, as $F_W = \frac{v_{slip}}{cv_r}$. This allows slips for turning while preventing stationary vehicles from slipping down slopes. wheel_slip is not included in core ODE but as a Gazebo plugin written in ODE. The plugin works by *dynamically* updating the ODE slip parameter, as used statically in Slip friction (FDS) above.

2 Proposed Protocol

Contact softness First, tune the number and location of wheel-surface contacts to give one reliable contact per wheel, using ODE parameter kp to govern the contact 'softness'. In ODE, friction is independent of contact area, so contacts can be modelled as single points. If kp is too soft, the robot can get two contacts per wheel or sink into the ground. Contacts should be in the center of the wheel, not towards either edge: real wheels usually have a curved profile which should be simulated; or using a very narrow simulated wheel can be a close approximation.

Coulomb parameters The limit of Coulomb friction is generally not desirable to hit as it represents skidding out of control. This makes it difficult to test safely with larger robots and so an arbitrary large value such as 10 or 1000 is appropriate for many applications. If it is really needed, options include running the robot into a fixed object until the wheels spin and measuring the force applied or measuring the braking distance for an emergency stop. Tables of μ for standard surface type pairs are widely available, ranging from 0.05 (teflon-teflon) to 1.2 (rubber-asphalt). The $fdir$ parameter needs to be set to ensure that the longitudinal and latitudinal parameters follow the orientation of the robot. Pragmatically, this is most easily checked by setting one slip parameter much larger than the other and tilting the world using the ODE gravity parameters.

wheel_slip setup this ODE plugin should be enabled, with normal force 1.0. The normal force is used as an extra multiplier to correct units but is not practically useful over and above the wheel slip constant as there is no capability to dynamically vary the force. The integration of the plugin to the SDF file is shown in the source code [3].

Step size tuning If the step size is too large, the response to large differences in long-lat slip is 'damped'. To tune step size, the lateral slip test below can be set up in simulation and the achieved ratio measured.

Longitudinal wheel_slip Initial odometric calibration should be done at the slowest practicable speed such that the wheelslip experienced by the robot is minimised. Subsequently, to calibrate the longitudinal wheel_slip, the robot is set to move a fixed distance (d_t) at a target velocity (v_r) as determined by its odometry. The acceleration, a, experienced by the robot is measured by the IMU to determine the force profile on the robot and the longitudinal wheelslip parameter (c_{long}) tuned such that the overall slip is consistent with the distance seen by the wheel odometry. The test is proposed as a step function between a lower start speed and a higher target speed such that numerical errors around low velocities are minimised. As the vehicle moves straight forward, acceleration is provided entirely by longitudinal friction of the wheels on the ground,

$$a = \frac{F}{m} = \frac{4(v_r - v)}{c_{long} v_r m}, \quad v_r = \frac{v}{1 - 4ac_{long}m}. \tag{1}$$

Acceleration can be numerically integrated to find velocity $v(t)$ and the c_{long} parameter tuned by search such that

$$d_t = \int \frac{v}{1 - 4ac_{long}m} dt. \tag{2}$$

Lateral wheel_slip This test involves the robot being spun on the spot given a constant target angular velocity ω_{target} and the achieved steady state angular velocity ω measured by counting the time taken to make n rotations. This provides a ratio between the latitudinal and longitudinal slip parameters. Care must be taken during the test that neither latitudinal nor longitudinal slip is in the Coulomb region where μ is dominant. This can be back-calculated by comparing the slip forces against μN after the test and rerunning with a lower ω_t if required. For the tested robot model, the centre of mass was coincident with the centre of the wheelbase but a correction term would be needed if this was offset.

Figure 1 shows the force balance for the test. Resolving moments about the robot centre, $bF_{long} = lF_{lat}$,

$$\Rightarrow F_{long} = \frac{b\omega_{target} - b\omega}{b\omega_{target} c_{long}}, \quad F_{lat} = \frac{l\omega}{b\omega_{target} c_{lat}} \Rightarrow \frac{c_{lat}}{c_{long}} = \frac{l^2\omega}{b^2(\omega_{target} - \omega)}. \tag{3}$$

3 Validation

A Pioneer P3AT simulation with known parameters was used to generate data using the protocol: $c_{long} = 0.005, c_{lat} = 0.05, \omega_{target} = 1.0, CFM = 0, ERP = 0.2, Stepsize = 0.0001$. We then attempted to recover c_{long}, c_{lat} from the data. The cone friction model was used as conceptually more accurate. Both directions' μ were set to 10.

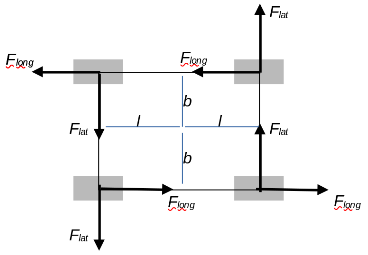

Fig. 1: Force balance for a skid steer robot spinning on the spot.

Rotation test The ratio $\frac{c_{lat}}{c_{long}}$ was calculated from the data by eqn. 3, as 9.998 (0.2% error). The simulation was not exactly aligned due to the finite step size but this result shows that the model and the equations are consistent. For physical experiments, a small range of ω would be tested to ensure the assumed slip region linearity was plausible.

Longitudinal test target velocity of 1.0m/s and test duration 1s were used. Torque limit was reduced to limit vehicle acceleration and prevent 'wheelies', and the instantaneous accelerations and velocities calculated numerically from reported positions. The 'encoder' distance was determined using the wheel angular velocities and the velocity and acceleration signals calculated numerically from 20Hz position measurements. A maximum limit for c_{long} was determined according to $c_{long} < \frac{4}{a_{max}m}$ (to avoid div0 errors) and bisection used to find the matching c_{long}. This test was repeated for a range of c_{long} values but consistently underestimated c_{long} (the contacts looked 'stiffer' than they were supposed to) and plots of acceleration were very noisy. This is thought to be due to the use of numerical differentiation. This effect would not be present in real life tests as the acceleration could be measured directly with an IMU. However, the measurement is fundamentally challenging because the amount of slip demonstrated over the length of a lab floor would be expected to be small. In cases where the measurement is not repeatable, it may be more appropriate to pick a stiff c_{long} based on IMU resolution (i.e. 'based on these quick tests, we know the tyres are at least this grippy').

The validation suggests that the new protocol may work to reduce the time currently spent by modellers performing manual search for realistic parameters.

References

1. S. Khaleghian, A. Emami, and S. Taheri, "A technical survey on tire-road friction estimation," *Friction*, vol. 5, pp. 123–146, 2017.
2. H. B. Pacejka, "Theory of Steady-State Slip Force and Moment Generation," in *Tire and Vehicle Dynamics.* Butterworth-Heinemann, 2012, pp. 87–147.
3. "osrf/gazebo," Jul. 2022, original-date: 2020-04-19T04:26:21Z. [Online]. Available: https://github.com/osrf/gazebo/blob/gazebo11/plugins/WheelSlipPlugin.hh