Drivetrain Simulator

AMB Calculator

The drivetrain simulator is used to find the proper gear ratio for a robot's drivetrain. By choosing the desired distance to traverse, you can optimize the gear ratio to give the best combination of acceleration and top speed. This simulation takes into account a number of factors, including wheel slip, current limits, voltage ramps, and voltage sag. It also features a "Predictive Stop" mode, which attempts to decelerate the robot before it hits the target distance in order to arrive at the target at zero velocity.

Simulation Calculations

The simulation runs one timestep at a time, starting from time $t = 0$ at position $x = 0$ and velocity $v = 0$. We will start the simulation by applying the full battery voltage in the forward direction.

Between the battery and motor, there is voltage sag due to high current draw through a system resistance defined as R_{sys}. To model this, when the battery is connected, resistive losses are subtracted from the magnitude of the battery voltage to get the voltage at the motor. Since we do not yet know what current will be drawn, we use the current drawn in the previous timestep I_{prev} for the calculation.

$$V_{motor} = \pm V_{batt} - I_{prev} \cdot R_{sys} \cdot \text{sign}\left(V_{batt}\right) \tag{1}$$

Rather than use the FRC motor parameters $\omega_f,\ T_s,\ I_s,\ I_f$, we will use the more standard DC motor parameters: torque constant k_m, back-EMF constant k_e, and armature resistance R. These can be calculated from the FRC motor parameters with the formulas:

$$k_m = \frac{T_s}{I_s - I_f}\ ; \qquad k_e = \frac{12}{\omega_f}\ ; \qquad R = \frac{12}{I_s - I_f} \tag{2}$$

We can then use the standard DC motor equations for applied voltage V, output speed ω, current draw I, and motor torque T_{motor}:

$$V = I \cdot R + k_e \cdot \omega\ ; \qquad T_{motor} = k_m \cdot (I - I_f) \tag{3}$$

The rotational output speed of the motor is equal to the robot's velocity divided by the wheel radius r, and multiplied by the gear ratio G. So we can calculate the total current draw across all n motors as:

$$I = \frac{n}{R}\left(V - k_e \cdot \frac{v \cdot G}{r}\right) \tag{4}$$

We can apply the current limit by limiting the magnitude of this calculated current to be less than the desired value. In reality, this is done through complex algorithms to change the applied voltage in order to give the proper current draw. For the purposes of this simulation, we will assume the current is properly limited so $I \leq I_{limit}$.

With this current, we can use the second DC motor equation to calculate the torque generated by the motor. From there, the torque at the wheels T is the motor torque multiplied by the gear ratio, minus parasitic losses that scale linearly with the robot speed (an approximation, but good enough for the purposes of this simulation):

$$T = T_{motor} \cdot G \cdot \eta - T_s\left(1 - \eta\right) \cdot \frac{v}{v_{max}} \tag{5}$$

In order to account for wheel slip, we calculate the maximum torque that can be transferred through static and kinetic friction. This is the friction force multiplied by the wheel radius:

$$T_{slip_s} = \mu_s \cdot mg \cdot (\%W) \cdot r \; ; \qquad T_{slip_k} = \mu_k \cdot mg \cdot (\%W) \cdot r \tag{6}$$

At $t = 0$ the wheels are not slipping. If $T > T_{slip_s}$, the wheels begin to slip. If $T < T_{slip_k}$, the wheels stop slipping. If the wheels are slipping, the magnitude of the torque at the wheels is limited to T_{slip_k}, and we work backwards to calculate the associated current draw I.

With the limited torque at the wheel, we can calculate the net force on the robot, and therefore the robot's acceleration a:

$$F = \frac{T}{r} \qquad \Longrightarrow \qquad a = \frac{F}{m} \tag{7}$$

And then we can apply that acceleration to change the robot's velocity and position:

$$v = v_{prev} + a \cdot dt \qquad \Longrightarrow \qquad x = x_{prev} + v \cdot dt + \tfrac{1}{2} a \cdot dt^2 \tag{8}$$

Stopping Calculations

At the end of each timestep, we check if our conditions for stopping the simulation have been met, and if not how much voltage we should apply in the next timestep.

For stopping type "No Stop", the simulation is done when the robot reaches the sprint distance, $x \geq x_{sprint}$. For type "Stop After", the simulation is done when $x \geq x_{sprint}$ and the robot has come to rest, $v = 0$. For type "Predictive", the robot does not need to reach the sprint distance but it does need to have passed the stopping point (to be calculated shortly) and come to rest.

The stopping point x_{stop} is the distance at which the robot stops accelerating and begins decelerating, which is calculated based on the stopping type. If the stopping type is "No Stop" or "Stop After", the stopping point is the desired sprint distance, $x_{stop} = x_{sprint}$. If the stopping type is "Predictive", the stopping point is moved forward so that the robot should come to rest at the sprint distance. We will assume the deceleration a_{stop} is constant, which is not really the case but will give us a close enough result. Then from basic motion equations:

$$0 = v_f^2 = v_i^2 + 2a_{stop} \cdot \Delta x \implies x_{stop} = x_{sprint} - \frac{v^2}{2a_{stop}} \tag{9}$$

For "Brake" and "Reverse" stopping methods, we assume the wheels are slipping for the entirety of the deceleration, so $a_{stop} = -\frac{1}{m} F_{slip_k} = \mu_k g \cdot (\%W)$. For "Coast", the average deceleration is equal to approximately half of the maximum friction losses, $a_{stop} \approx -\frac{T_s(1-\eta)}{2mr}$.

If the simulation is not finished, we will calculate what voltage to apply to the system on the next timestep. If the robot has not yet reached its stopping point ($x < x_{stop}$), it is given full battery voltage. If it has reached the stopping point, the applied voltage is based on the selected stopping method. For "Coast", the battery is disconnected and no voltage is applied (i.e. the motor voltage is not constrained). For "Brake", the motor leads are "shorted", and the lead-to-lead voltage is forced to zero. For "Reverse", the full battery voltage is applied in reverse.

The calculations for this simulation are based in part on work done by Jesse Knight and FRC team 1885.