

FRC Drivetrain Characterization

Noah Gleason and Eli Barnett

noah.gleason1@gmail.com

emichaelbarnett@gmail.com

The Blair Robot Project: FRC Team 449

November 2017

1 Introduction

In this paper, we discuss a method of characterizing motor, gearing, and wheel (“drive”) systems on FRC robots. We present data and analysis that strongly suggest that the behavior of FRC drive systems is quite rigorously linear, and that several commonly-used techniques for drive characterization in FRC are ill-founded. We further discuss potential theoretical implications for FRC drivetrain modeling, and practical applications for determining accurate control-loop feedforwards from characterization data.

For the first part of this paper, we will discuss the theoretical behavior of a robot drive under several simplifying assumptions, including, importantly, the absence of friction. We will then discuss the violations of these assumptions in actual FRC robot drives, appropriate modifications to deal with these violations, and finally conclude with practical instructions for implementing the resulting drive characterization in robot control.

2 Theoretical Foundations

2.1 The Voltage-Balance Equation

The voltage balance equation¹ for a permanent-magnet DC motor is given by

$$V_{app} = k\omega + IR \tag{1}$$

where V_{app} is the voltage applied across the motor, k is some constant, ω is the angular velocity of the rotor, I is the current through the motor windings, and R is the resistance across the motor windings. One can think of this equation as “partitioning” the applied voltage to the motor in two:

$$V_{app} = V_{emf} + V_{windings} \tag{2}$$

where

$$V_{emf} = k\omega \tag{3}$$

and

$$V_{windings} = IR \tag{4}$$

¹All quantities in this paper have SI units, with the exception of distance (and all derived quantities such as velocity, acceleration, etc) which is given in units of feet for sake of familiarity to most FRC teams.

Here, V_{emf} is the so-called “back-EMF,” and can be thought of as the result of the rotor’s motion causing the DC motor to act as a generator. Our other term, $V_{windings}$, corresponds to the voltage drop across the motor windings.

We can further think of the V_{emf} as the voltage required to keep the motor spinning at a desired constant speed in the absence of any load, or, in other words, the “portion of voltage” corresponding to the motor’s velocity. Similarly, we can think of $V_{windings}$ as the “portion of voltage” corresponding to the motor’s output torque, since I is proportional to the strength of the field generated by the motor windings, and thus to the torque generated by the motor.

We now consider the implications for a robot drive. Let us consider a hypothetical “tank-drive” robot driven by one motor per side of the drive, with constant gearing (the case of multiple motors per side is fundamentally equivalent, so long as the motors are geared together; we can consider the group of motors as a single motor). Let us further assume that the wheels do not “slip” on the ground, and that each side of the robot moves in unison.

By observing that, under these conditions, the rotor speed ω is proportional to robot velocity, and the output torque is proportional to robot acceleration, we can (by judiciously “absorbing” constants) rewrite our earlier equations in terms of robot velocity and acceleration:

$$V_{emf} = k_v \cdot \text{velocity} \quad (5)$$

and

$$V_{windings} = k_a \cdot \text{acceleration} \quad (6)$$

thus, we obtain

$$V_{app} = k_v \cdot \text{velocity} + k_a \cdot \text{acceleration} \quad (7)$$

A rather important point is now immediately apparent: the voltage applied to our robot’s drive can be partitioned into a portion corresponding to the robot’s velocity, and a portion corresponding to the robot’s acceleration. Characterizing the behavior of a robot drive, then, becomes a matter of determining the values of our two constants, k_v and k_a .

2.2 Theoretical Determination of k_v and k_a

In order to get a clearer sense for the meanings of our constants, k_v and k_a , we now consider two hypothetical situations.

Firstly, we consider the case of a robot at steady-state, with full voltage V_{max} applied to the motors. Since the robot is at steady state, we know that acceleration is zero, and all of the applied voltage must go towards maintaining the robot velocity (or, in other words, counteracting the “back-EMF”), corresponding to the fastest possible robot velocity, which we will call velocity_{max} . Thus we have

$$V_{max} = k_v \cdot \text{velocity}_{max} \quad (8)$$

and

$$k_v = \frac{V_{max}}{\text{velocity}_{max}} \quad (9)$$

Similarly, we consider the case of a robot at rest, again with full voltage V_{max} applied to the drive motors. Since the robot is at rest, we know that velocity is zero, and thus all of the applied voltage goes towards accelerating the robot. This is the condition of the robot with the motors at stall, and corresponds to the fastest possible robot acceleration, which we will call $\text{acceleration}_{max}$. Thus we have

$$V_{max} = k_a \cdot \text{acceleration}_{max} \quad (10)$$

and

$$k_a = \frac{V_{max}}{\text{acceleration}_{max}} \quad (11)$$

We now know enough to obtain a rough theoretical guess for the values of k_v and k_a , as V_{max} is known and $velocity_{max}$ and $acceleration_{max}$ can be calculated from the motor free-speed and stall-torque, respectively, in relatively-straightforward fashion: Let ω_{free} be the free speed of the motor, τ_{stall} be the stall torque of the motor, n be the total number of drive motors, m_{robot} be the mass of the robot, and d_{wheels} be the diameter of the robot’s wheels, and $r_{gearing}$ be the total gear reduction between the motors and the wheels. Then, it is easy to see that

$$velocity_{max} = \frac{\omega_{free} \cdot \pi \cdot d_{wheels}}{r_{gearing}} \quad (12)$$

and, neglecting the effect of moment-of-inertia of the drivetrain,

$$acceleration_{max} = \frac{2 \cdot n \cdot \tau_{stall} \cdot r_{gearing}}{d_{wheels} \cdot m_{robot}} \quad (13)$$

However, while these formulations for k_v and k_a are useful in a theoretical sense, they are difficult to employ in practice, mainly because both $velocity_{max}$ and $acceleration_{max}$ are somewhat difficult to empirically measure: due to various effects such as friction, wheel slip, and battery “voltage sag,” these quantities do not necessarily correspond to the measured maximum robot velocity or acceleration. Additionally, as we will see in the next section, a small modification must be made to equation 7 when the effect of friction is considered.

3 Drive Behavior in Practice

Our analysis thus far provides a good foundation for the characterization of FRC robot drives. However, as noted, several simplifying assumptions were made, including: straight robot motion, lack of wheel slip, and, most importantly, the absence of friction.

3.1 Accounting for Friction

Practical application of the above equations is rendered somewhat more complicated by considerations of reality. In particular, the presence of friction proves problematic, and demands special consideration. However, our testing reveals that the frictional effects present in a typical FRC drive are overwhelmingly linear, and thus accurately accounting for friction is surprisingly simple.

The reader is likely familiar with some of the effects of friction on robot drives. For example, a max robot speed calculated directly from the robot’s free-speed with no correction for frictional losses is well-known to be unreachable in practice. Likewise, for those teams that have measured robot acceleration, it is very clear that the “theoretical” maximum acceleration as calculated from the robot’s stall torque is far greater than anything that can actually be achieved. The use of frictional “fudge-factors” to account for these issues is commonplace, usually in the form of a simple constant multiplier (a common value is in the neighborhood of .8) applied to to all relevant quantities. As we will demonstrate later in the paper, such naive approaches are not correct in all cases, and can lead to poor characterization of robot behavior.

To gain insight into the actual effects of friction on robot, we conducted a number of tests on our test robot. The test robot in question weighed approximately 110 lbs, and due to some notable issues (including minor dragging of the belly-pan on the ground and an old, heavily-used drivetrain dating back to 2014) had fairly substantial frictional losses in the drive train. Our test robot had a 6-CIM West-Coast drive, composed of WCP single-speed 3-CIM gearboxes driving 4” Colson wheels, worn to 3.8” effective diameter, with a gear ratio of 6.1:1, with power transmitted to the outer wheels via 9mm-wide 5mm-pitch HTD belts. The robot was driven on a carpet highly similar to that used on FRC fields.

3.2 The Steady-State Voltage-Speed Curve and Determination of k_v

The equations for drive behavior in the absence of friction describe a linear - in fact, proportional - steady-state voltage-speed curve. Suspecting that friction may result in a nonlinear real-world voltage-speed curve, a test was conducted in which the test robot was run to steady-state speed at an array of positive applied voltages. The resulting steady-state speed was measured by drive encoders. To account for “voltage sag” from the battery and robot wiring, the actual voltage applied to the motors was directly measured via functionality offered by its Talon SRX motor controllers. The resulting voltage-speed curve is shown below (for the sake of brevity, only the curve for the left side of the robot’s drive is presented - as the robot did not track entirely straight, the curve for the right side of the drive differed slightly):

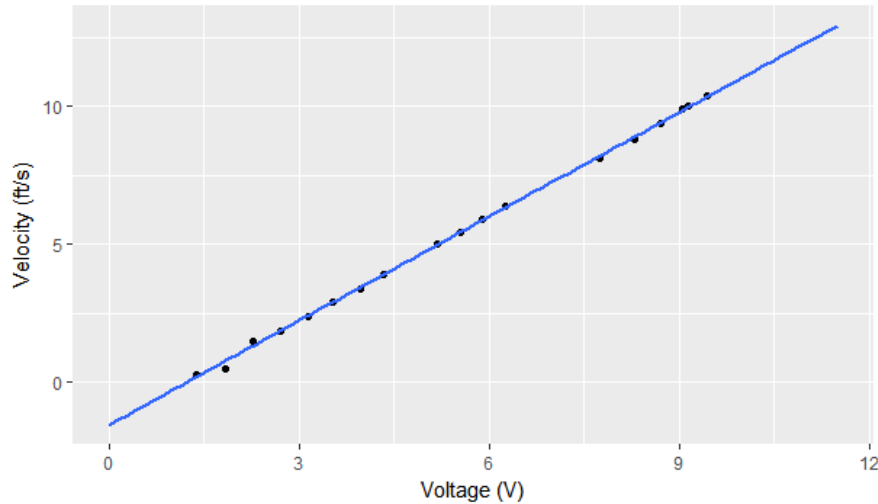


Figure 1: The steady-state velocity versus voltage curve for the left side of the test robot

We can see several important features from this graph. Firstly, and most importantly, the steady-state voltage-speed curve is extremely linear, indicating that steady-state frictional effects are linear with robot velocity. Accordingly, the steady-state frictional effects can be partitioned into two parts: constant friction, and friction proportional to velocity.

Secondly, we note that the purely proportional relationship between voltage and velocity suggested by equation 7 is quite obviously inaccurate; the real-world voltage-speed curve has a nonzero intercept. This is unsurprising - it is well-known that FRC robot drives have a voltage “dead-zone” around zero within which the torque generated by the motors is insufficient to overcome frictional losses in the drive.

The presence of a nonzero intercept in the steady-state voltage-speed curve, combined with the verified linearity of the behavior, suggests a simple amendment to equation 7:

$$V_{app} = k_v \cdot \text{velocity} + k_a \cdot \text{acceleration} + V_{intercept} \quad (14)$$

Here, $V_{intercept}$ is the x-intercept of the voltage-speed curve above; for our data, the value is ~ 1.26 V. One can think of $V_{intercept}$ as the voltage required to generate enough torque to overcome the constant steady-state frictional effects.

Observe that this test offers us a simple empirical determination of k_v : we can take k_v to be the inverse of the slope of the voltage-speed curve. Doing this “absorbs” any steady-state frictional effects proportional to velocity into the value of k_v . Given the observed linearity of the robot behavior, this

together with the addition of the intercept accurately accounts for nearly all steady-state frictional effects.

We can further compare our empirical value of k_v to the “theoretical” value calculated from the motor’s free speed. Surprisingly, our measured value for this test is extremely close to the theoretical value; we obtained an empirical k_v of $\sim 0.81 \text{ V s ft}^{-1}$, as opposed to a theoretical value of $\sim 0.83 \text{ V s ft}^{-1}$. As CIM free speeds are only guaranteed to $\pm 10\%$, the difference is probably not significant.

It is interesting to consider the close agreement between the empirical and theoretical values of k_v despite the common knowledge that FRC robots do not actually reach their “theoretical” top speeds. There are two main reasons for this. Firstly, it is impossible to ever actually apply maximum battery voltage to the motors, due to “voltage sag” caused by the internal resistance of the battery. This can be seen in figure 1; despite our commanded test voltages ranging all the way up to the ostensible maximum of 12 V, the maximum voltage actually measured reaching the motors was $\sim 9.5 \text{ V}$. Secondly, the voltage actually available to cause steady-state robot speed can be seen from equation 14 to be $V_{app} - V_{intercept}$; thus, additional top speed is lost to steady-state friction. Note, however, that the steady-state frictional losses are overwhelmingly constant, not proportional, since the empirical value of k_v agrees so closely with theory: we can see from this that the common approach of multiplying the theoretical top-speed by a corrective frictional “fudge-factor” to obtain an estimated practical top-speed is incorrect.

3.3 Dynamic Robot Movement and Determination of k_a

Our real-world analysis thus far has accounted for steady-state robot behavior, and corresponding provided values for $V_{intercept}$ and k_v . It remains to investigate dynamic robot behavior - that is, what the robot does when it is changing speeds - and determine a value for k_a .

From equation 14, we can see:

$$k_a \cdot \text{acceleration} = v_{app} - (k_v \cdot \text{velocity} + V_{intercept}) \quad (15)$$

This naturally suggests a procedure for the determination of k_a , given our earlier empirically-determined values for k_v and $V_{intercept}$. We can think of $v_{app} - (k_v \cdot \text{velocity} + V_{intercept})$ as the “portion of voltage” causing robot acceleration. Our equation again tells us that, like the steady-state voltage-speed curve, in the absence of friction, the relationship between this and robot acceleration should be linear, with slope k_a .

As in the previous section, we suspected that the real-world result may be nonlinear. In order to test this, we ran the test robot with a constant applied voltage (60% of max voltage was chosen; we did not want to slip the wheels, which would have invalidated the test). As in previous tests, voltage applied to the motors and drive velocity were both logged. Acceleration was calculated as the slope of the secant line across four time-slices of velocity data; this was found to be necessary to reduce the very large amount of noise. Acceleration measurements were also made with an accelerometer (NavX MXP), and again found to be too noisy. The collected data were also “trimmed” to remove an initial period of “acceleration ramping” caused by motor inductance, which we do not attempt to describe in our analysis. The resulting plot of acceleration versus “acceleration-causing voltage” (as described earlier, given by $v_{app} - (k_v \cdot \text{velocity} + V_{intercept})$ with the values of k_v and $v_{intercept}$ obtained from our steady-state test) is shown below, along with the corresponding plot for robot velocity versus time:

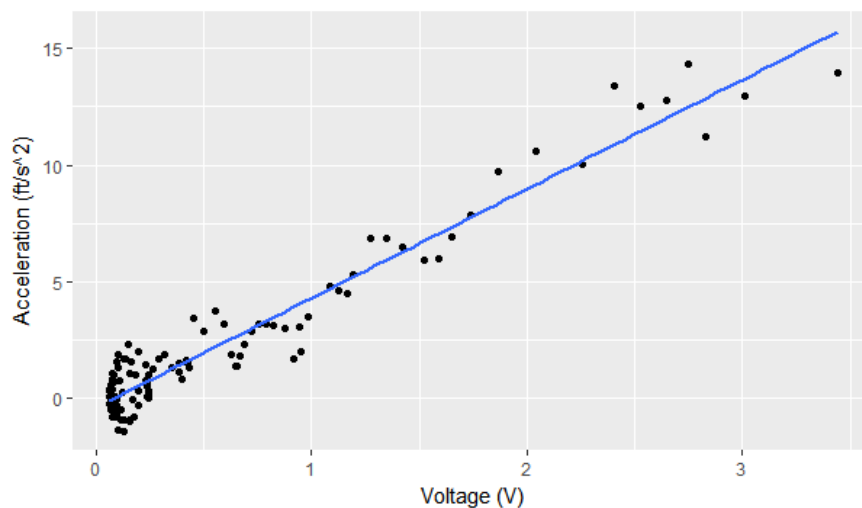


Figure 2: The acceleration versus voltage curve for a robot, measured via on-ground step voltage tests.

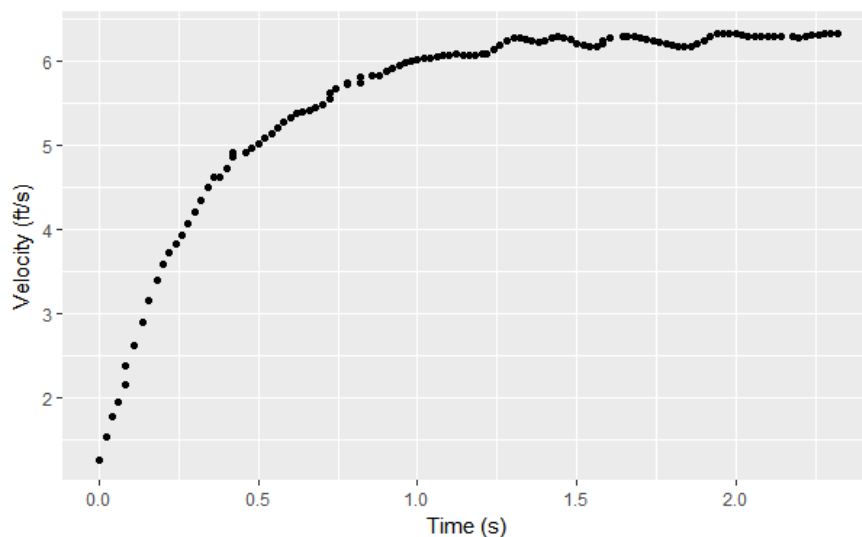


Figure 3: Velocity of robot versus time during test.

Unsurprisingly, the curve is much noisier ²; however, no serious nonlinearity seems to be present. Additionally, the intercept of the regression line is very nearly zero, corresponding to approximately 0.08 V. Were our steady-state characterization perfect, this intercept ought to be zero, so the small value is reassuring and indicates agreement between the steady-state and dynamic tests. Additional tests at different voltages yielded similar results. ³

As in the previous test, we can obtain an empirical value for k_a by simply taking the reciprocal of

²A close observation of the velocity-versus-time plot shows that the data are not quite evenly-spaced in time (in fact, some values of time have two data points!); this is due to a combination of inconsistent roboRIO loop execution time and the use of a parallel logging routine, and logging time from a global robot clock value that is only updated upon main loop execution. While mildly inconvenient, it does not seriously impact the accuracy of the data.

³Current was also logged during the acceleration tests, and was found (as predicted by equation 7) to be essentially proportional to $V_{app} - k_v \cdot \text{velocity}$, with a linear regression yielding an r^2 of $\sim .993$ and an intercept of ~ 0.3 A

the slope of the regression line. For the data in the plot above, this yields a value of $\sim 0.21 \text{ V s}^2 \text{ ft}^{-1}$. We can again compute a theoretical value for k_a , as well, this time from the stall torque; this yields a value of $\sim 0.1 \text{ V s}^2 \text{ ft}^{-1}$.

Here, unlike in the steady-state test, we have a significant discrepancy between the empirically-measured value and the theoretically-calculated one - more than a factor of two! To understand the cause of this discrepancy, remember that we are “absorbing” proportional friction effects into our measured values for k_v and k_a . In the steady-state case, there appear to be few friction effects that vary significantly with robot velocity. However, the dynamic case is not analogous: friction between the gears will clearly vary with acceleration, as the normal force between gear teeth is proportional to the output torque of the motors, and thus to robot acceleration. Thus, the amount by which our empirical value of k_a is inflated from the theoretical is representative of the torsional loss from the motors to the wheels as a result of frictional effects that vary with load. From these data, it appears that the drive of our test robot loses over half of its motor torque to such losses.

As we can see from the linearity of our plot, these effects are (thankfully) quite linear, and thus we do not need to introduce any further changes to equation 14 to account for them.

On a final note, the astute reader may have noticed that running two separate linear regressions is somewhat unnecessary - we may simply combine the results of our tests and run the multivariate linear regression suggested by equation 14 to obtain values for k_v , k_a , and $V_{intercept}$ all at once. Doing this yields similar results to running sequential univariate linear regressions as was done above ⁴.

4 Practical Implementation

We now, finally, discuss what a FRC team might want to do with the information presented above, and how to do it.

4.1 Using the Equations: Open-Loop Control and Feedforward

With our empirical determinations of the unknown constants, equation 14 can now be implemented for actual robot control. In particular, equation 14 immediately gives us a very good approximation of the voltage we must apply to our robot’s motors to achieve a desired velocity and acceleration.

Most teams are likely more interested in achieving velocities rather than accelerations. In this case, the acceleration term of equation 14 can simply be omitted. This yields an equation similar to most open-loop control; the major difference introduced by our new knowledge is the importance of $V_{intercept}$. Any attempt to control robot velocity which omits $V_{intercept}$ is akin to attempting to model the curve in figure 1 with a line constrained to pass through the origin. Thus, the standard practice of simply sending motors a signal proportional to the desired velocity is clearly flawed.

To correct for this, we simply need to add $V_{intercept}$ to our voltage outputs. However, care must be taken when implementing this. The sign of $V_{intercept}$ clearly must change depending on which direction is commanded. Additionally, the addition of a nonzero intercept imposes a nonzero “minimum value” to our motor output; in order to avoid rapidly-fluctuating motor outputs in the region near zero (where the sign of the intercept will change), it is probably advisable to implement a “deadband” in which no output is given.

Note also that, as described earlier, this characterization suggests that the typical method of determining robot top speed in FRC by multiplying the theoretical top speed by a frictional “fudge factor” is incorrect, and unlikely to yield an accurate value. A better approach, if a theoretical guess at the value for the robot top speed is needed, would be to make a guess at the value of $V_{intercept}$; from this, the theoretical value of k_v , the current curve of the motor, and the internal resistance of the battery, the top speed of the robot can be easily calculated (the details of this calculation are left as

⁴An even-more astute reader might note that our “sequential” linear regression approach is unable to detect an “interaction” effect between velocity and acceleration; more-thorough investigation with an enlarged data set from several tests does not reveal any such interaction.

an exercise to the reader). Our analysis does suggest, however, that a multiplicative frictional fudge factor is indeed appropriate when considering the output torque, rather than the speed, of a drive. We hope that, if teams adopt this characterization methodology, reasonable guesses for these quantities will become known as more data is obtained from actual robots.

The full form of equation 14 (including the acceleration term) becomes useful in more-nuanced applications, such as motion-profile following, in which a meaningful acceleration setpoint can be determined. For such applications, equation 14 provides an ideal feedforward formulation. A common complaint around motion-profile following is that, when implementing feedforward of the form of equation 7 (without $V_{intercept}$), it is difficult to obtain a set of feedforward gains (k_v and k_a) that works well across a wide range of potential cruise velocities; our testing indicates that, with feedforward implemented following the form of equation 14 with constants determined empirically as we have described, open-loop following of linear trapezoidal-acceleration motion profiles can be made accurate within $\sim 2\%$ across a wide range of cruise speeds. This greatly reduces the amount of corrective work that must be done by the feedback portion of the profile-follower.

A number of caveats apply to our suggested methodology, however. Firstly, it is absolutely crucial that some step be taken to account for “voltage sag” due to the internal resistance of the battery and resistance of the robot’s wiring. The Talon SRX motor controller offers a “closed-loop voltage compensation” option that does a fine job of accomplishing this; battery voltage monitoring via the PDP can also be used to apply the necessary correction.

By the same token, when empirically determining k_v and k_a , it is similarly important to measure the actual voltage being applied to the motors, as due to “voltage sag” this can vary significantly from the commanded voltage.

The two sides of a robot’s drive may not behave identically (this can easily be seen as a failure of the robot to track straight when each side is given an identical voltage), due to differences in friction or asymmetry in the operation of the motors (many brushed DC motors are designed to be more-efficient in one direction than the other; there is, as far as we know, no consensus in the FRC community as to whether this is true for CIMs). Ideally, one would like to somehow test each side separately, but as it is not possible to physically decouple the left of a robot from the right of a robot (by any method we know of), this is not possible - we have found that, in practice, it is enough to simply run the calculations described above separately for each side of the drive, obtaining a separate set of constants for each.

4.2 Quasi-Static Determination of k_v

Determining the steady-state voltage-speed curve can be labor-intensive. Rather than running a large number of constant-voltage trials, each to steady-state, one can take a “short-cut:” a quasi-static test can be run, instead.

In the quasi-static test, the voltage given to the drive is slowly ramped upwards. If the ramping rate is slow enough, the acceleration term in equation 14 is negligible, and can be ignored; the data needed to calculate the voltage-speed curve can thus be obtained in a single test.

A few weeks after our initial steady-state tests, we ran a quasi-static test on our practice bot to determine the accuracy of the method and an appropriate ramp rate. After some iteration, we found that a ramp rate of 0.25 V s^{-1} worked very well and produced results near-identical to those of our steady-state trials. It should be noted, however, that much more space is needed to reach full speed when slowly ramping the voltage; we were unable to reach more than approximately half of max speed before reaching the end of our carpet during the quasi-static test. However, given the utter lack of nonlinearity in the steady-state voltage-speed curve apparent in our testing, this is likely not an issue. The steady-state voltage-speed curve from the quasi-static test is shown below:

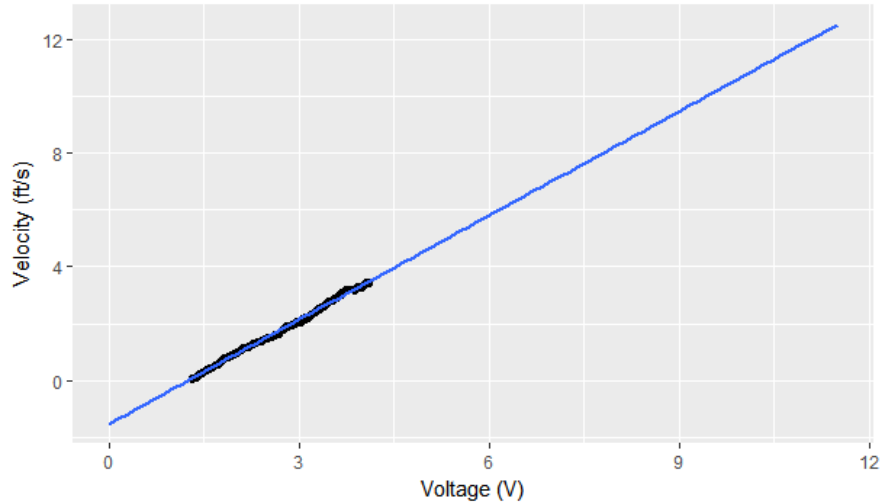


Figure 4: The velocity versus voltage curve for a robot, measured via an on-ground quasi-static test.

4.3 Implementation with Existing FRC Hardware/Libraries

Unfortunately, as of the time of writing this whitepaper, support for custom feedforward curves in FRC is sparse. Most PID implementations offered to FRC teams offer feedforwards only of the form $k_v \cdot \text{velocity}$, which we have noted to be substantially inaccurate. Fortunately, additional support is on the way, and there exist some “hacks” to allow implementation on some currently-available tools.

The WPILib PIDController object constrains feedforward to the form $k_v \cdot \text{velocity}$. While it is messy/inconvenient to get around this for the PIDController object itself (we will not discuss potential solutions here, though they are not too hard to think of), WPILib offers two convenience wrappers for the class in the form of the PIDSubsystem and PIDCommand objects. Both include a “usePIDOutput()” method, which can be easily modified to do anything that the user desires with the output of the feedback loop, including adding a custom feedforward to it.

The TalonSRX currently does not offer custom feedforward support in any of its closed-loop modes, though the next version of the firmware is slated to include a “throttle bump” feature which will allow for custom feedforward implementations. However, it is possible to “hack” the motion-profile control mode to support custom feedforward by adding the desired feedforward to the velocity setpoint, as it is only used by the controller for the feedforward (and not for feedback). As the controller multiplies the velocity setpoint by the user-specified feedforward gain, care must be taken to ensure that the values are properly-scaled to result in the correct voltage after computation by the controller. One approach to this is to set the Talon feedforward gain to k_v and to divide both $V_{intercept}$ and k_a by k_v prior to adding them to the velocity setpoint; another is to set the Talon feedforward gain to $1023/V_{max}$ (value due to the Talon’s implicit units for the feedforward gain) and to replace the velocity setpoint with the desired feedforward voltage.

The TalonSRX offers a “minimum output” feature in closed-loop control modes, ostensibly to serve a similar purpose to $V_{intercept}$. This should not be used, as it does not actually add the “minimum output” to all output voltages (but rather simply “promotes” voltages that are too small to the minimum), and thus does not result in a correct voltage-speed curve.

5 Conclusion

The behavior of drivetrains, and most likely other FRC mechanisms using brushed DC motors, can be accurately characterized by three relatively easily-measured constants, as described by equation 14. The linearity of motor behavior even in the presence of frictional losses appears to be confirmed by experiment. The authors hope that WPILib, and other FRC libraries, will add support for motor control informed by this information in the future.

6 Data

[Static voltage vs velocity data](#)

[Voltage vs acceleration data](#)

[Quasi-static voltage vs velocity data](#)