

---

# Anticogging: Torque Ripple Suppression, Modeling, and Parameter Selection

The International Journal of Robotics Research  
2016, Vol. 35(1-3):148–160  
©The Author(s) 2015  
Reprints and permission:  
sagepub.co.uk/journalsPermissions.nav  
DOI: 10.1177/0278364915599045  
ijr.sagepub.com/

Matthew Piccoli and Mark Yim

## Abstract

Smooth motion is critical to some robotic applications such as haptics or those requiring high precision force control. These systems are often direct-drive, so any torque ripple in the motor output must be minimal. Unfortunately, low torque ripple motors are expensive. Low cost brushless DC motors are becoming more prevalent, especially from the hobby RC community. These motors often have the required high torque density; however, they also have significant torque ripple. This paper presents a low cost method for *anticogging* - the compensation of cogging torque. While other methods exist to compensate for current-based torque ripple (mutual or reluctance torque), none have addressed cogging torque, except by adding expensive force sensors. This paper presents two methods that use a position sensor (already present for servo motors) to map cogging torque to rotor position. The map is played back according to position reported from the sensor to cancel the cogging torque. The design and testing of a low cost haptic arm using anticogging shows validation; however, the approach is much broader, and can be applied to any precision force application. A model of torque ripple sources are included as a function of PWM frequency to help choose the optimal PWM frequency to minimize torque ripple. Test results on eleven different motors show a removal of up to 88 % of torque ripple with no added cost in robotic servo applications, and in some cases having better performance than motors that are over nine times as expensive.

## Keywords

Motor control, motion control, torque ripple, cogging torque, calibration and identification, force and tactile sensing, force control, smart actuators

## Introduction

The goal of many robotics companies is to move away from the high price industrial markets toward low-cost consumer markets. When cost-reducing these systems, the motors are often the most expensive element. Achieving nearly an order of magnitude reduction in cost has the potential to enable new markets for robotics.

Robot designers often use permanent magnet synchronous motors (PMSM) such as brushless DC motors (BLDCM) and brushless AC motors (BLACM) when motor torque, speed or precision performance is a concern to robot operation. PMSMs exhibit high torque to weight and inertia ratios. Compared to their AC induction counterparts, they are more efficient and simpler to control. Unlike brushed motors, PMSMs do not require brushes to commute and can be made more reliable and cheaper to manufacture. However, the commutation cost and complexity is now pushed to external controllers. Advancements in computation and miniaturization in power electronics are outpacing advancements in electric motors, so PMSMs are becoming even more attractive from a cost standpoint.

Torque ripple is the unwanted periodic fluctuation in the motor torque as the output shaft rotates. This has been recognized as a problem in a variety of robot applications Liu and Goldenberg (1993) Newman and Patel (1991) Wallace and Taylor (1991). In haptic rendering it is especially troublesome, where direct-drive, high torque motors are

desirable and often essential Hayward and MacLean (2007). Transmissions, such as gear boxes, add non-linear torque variations that are difficult to model and compensate, making direct-drive favorable. PMSM's high torque capability allow them to be used direct-drive and would be ideal if not for torque ripple.

The recent growth in the electric hobby RC market (in particular flying vehicles) has provided a wide range of high torque density, low cost motors. For example, the Exceed RC Rocket 86MA10 motor is 1/8th the price of a Maxon EC45 261501, is smaller and has higher maximum torque, but has a peak to peak torque ripple of 16 N mm, over 440 % that of the Maxon. When a motor spins at high speeds, torque ripple creates high frequency speed fluctuations that generate sound and vibration. In haptic rendering, humans are sensitive to periodic motions especially higher frequencies, 40 Hz to 100 Hz. With good compensation for the ripple, these unwanted vibrations can be reduced, a quadrotor's motors make less noise, robotic arms have smooth motion, and haptic textures are rendered more accurately. At very

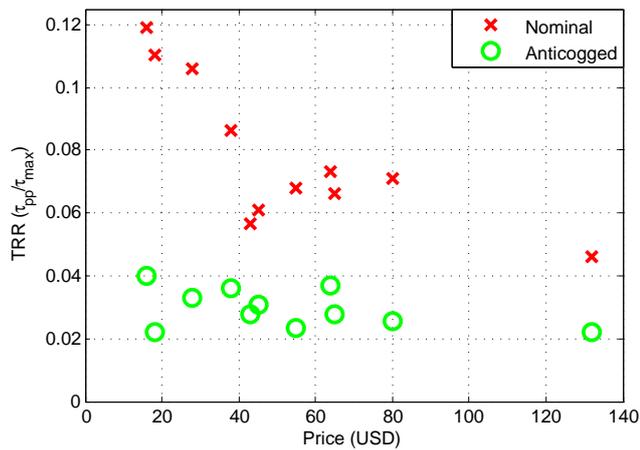
---

Department of Mechanical Engineering and Applied Mechanics,  
University of Pennsylvania, USA

### Corresponding author:

Matthew Piccoli, University of Pennsylvania, 220 South 33rd Street,  
Philadelphia, PA 19104-6315, USA.

Email: piccoli@seas.upenn.edu



**Figure 1.** Nominal and compensated torque ripple ratio vs. price for a set of motors of nominally same size.

low speeds, torque ripple can cause relatively large speed fluctuations, even causing the motor to stop or move in discrete increments. In servo control, precise positioning is impossible with a traditional proportional or proportional-integral controller due to the ripple's nonlinearity.

Figure 1 shows a graph of a sampling of hobby RC brushless motors measured by the authors along with some high performance ones (e.g. a Maxon EC 45, the right-most data point). A measure of torque ripple is shown in the figure as the torque ripple ratio ( $TRR$ , detailed later in Equation 13) which is the peak torque ripple, normalized with a motor's maximum torque. From this graph, one can see a correlation between lower priced motors and higher  $TRR$ .

This paper presents an anticogging method to compensate for cogging torque ripple that yields high performance from motors that are a fraction of the cost of inherently low torque ripple motors. By enabling low cost yet high performance motors, this work has the potential to transform the robotics industry by opening consumer markets for high performance robots that are practical and low cost enough for a wide range of useful tasks in the home.

### Types of Torque Ripple

There are four main types of torque ripple: mutual, reluctance, cogging, and friction.

*Mutual torque* is caused by the mutual interaction of the rotor's permanent magnets and the stator's currents Hung and Ding (1993) Park et al. (2000). In a PMSM, this is the primary source of torque production, having the largest DC component\*. A mismatch of the rotor's magnetic field and the stator's current waveform causes dips in the produced torque and contributes to torque ripple. Some sources of mutual torque ripple are driving a BLAC with a trapezoid or a BLDC with a sine wave, phase shifts or delays in the wave, low pwm resolution, and low pwm frequency.

*Reluctance torque* is a result of variance in the stator's self-inductance due to the rotor magnet saliency. The magnitude of reluctance torque is a function of current Petrovic et al. (2000). In an ideal BLACM (perfect sinusoidal back EMF and currents), reluctance torque does not exist or only contains a DC component. BLDCMs and non-ideal BLACMs contain reluctance torque ripple.

*Cogging torque*, also known as detent torque, comes from the rotor's permanent magnets' attraction to the salient portions of the stator Qian et al. (2004). It is not current-dependent and cannot be detected by a current sensor. It also has no DC component, and thus only contributes to torque ripple. Further explanation is in Extension 1.

*Friction torque* is not always axially symmetric, since bearings within the motor may contain eccentricities. These torque ripples are distinguishable from cogging torque by their once per mechanical revolution frequency and change in sign upon a change in direction.

BLACMs are intended to be driven off of AC mains, yielding a sinusoidal current waveform. BLDCMs are meant to be driven from a constant voltage source with a three phase inverter in a simple  $120^\circ$  commutation, yielding a trapezoidal current waveform. While the waveforms are similar enough between BLACMs and BLDCMs to interchange waveforms, it is not recommended as increased mutual torque ripple and efficiency losses will result. On the other hand, an inverter can mimic sinusoidal or trapezoidal waveforms using  $180^\circ$  commutation and pulse width modulation (PWM). Using this method, virtually any waveform within the supply limits can be generated, notably one that cancels all of the various types of torque ripple Le-Huy et al. (1986).

### Anticogging Background

Torque ripple minimization has been a topic of research for over 25 years. Many researchers have proposed finding an optimal current waveform offline using various methods and using a current controlled inverter to play back waveforms Hung and Ding (1993) Park et al. (2000) Le-Huy et al. (1986) Favre et al. (1993) Hanselman (1994). However, Aghili (2008) use current feedback, while Kim and Ha (1999) use speed feedback at low speeds for online estimation, and Holtz and Springob (1996) use both. In practice, speed control loops and estimation have limited success in minimizing torque ripple at higher speeds due to measurement delays, but Holtz and Springob (1996) show that it can be used at low speeds.

Petrovic et al. (2000) note that while cogging torque cannot be detected from current measurements, all forms of torque ripple are seen via added mechanical sensors. While a few prior works do mention the possibility of adding cogging torque suppression to their current based algorithms Hung and Ding (1993) Favre et al. (1993), none explore the specifics of finding the necessary waveform. Most reduction methods leave the suppression of cogging torque to the motor designers, typically by skewing the stator slots. In place of a speed loop, Qian et al. (2004) use an external force sensor as feedback to compensate for torque ripple at higher frequencies. This method suppresses all forms of torque ripple, but the required sensor could cost more than the motor itself.

Despite the progress in the above solutions, torque ripple minimization is not yet widely used in robotics. Torque ripple minimization is either incomplete when using current

\*When referring to DC components or DC signals, the authors are referring to the non-oscillating offset components in the frequency domain, rather than current.

sensing methods or is prohibitively expensive when using an external torque sensor. However, Armstrong (1988) has shown on robotic arms behind a gearbox that it is possible to measure torque ripple via position sensing by ramping current until an encoder indicates a position change as well as using acceleration feedback to model torque ripple. Unfortunately, observing accelerations may not work at high operational speeds, but monitoring speed and its ripple at low nominal speeds is comparatively simple and is possible with a position encoder. Data gathered at low speeds can be applied at high speeds open loop with notable results Armstrong (1988) Holtz and Springob (1996). An alternative method to monitoring speed ripple is to monitor position errors during position control. In the case of an unloaded motor during position control, cogging torque and friction are the primary torque perturbations. Therefore, position error under position control can be used to make a cogging torque map and friction torque map.

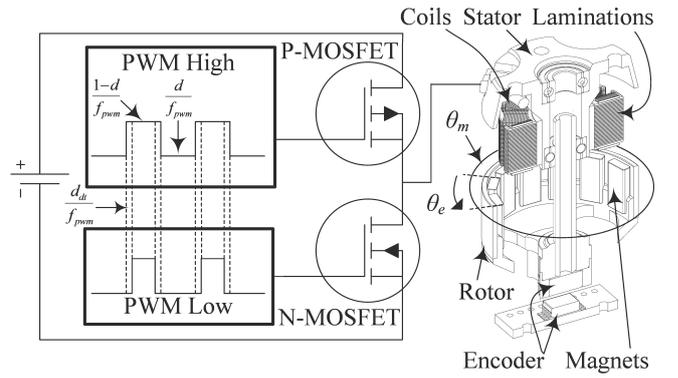
In our previous work Piccoli and Yim (2014), we showed that cogging torque waveforms can be estimated either by mapping speed fluctuations with respect to position or by mapping position error with respect to commanded position. Neither method requires more than an added position sensor which is already required for servo control, and both methods can capture all forms of torque ripple. The methods work with voltage control or current control with little change. One of the methods can be applied to sub-rotation intervals if the motor is constrained to certain positions, as in servo control of a joint. The results can be added to other algorithms to achieve complete torque ripple suppression Hung and Ding (1993) Favre et al. (1993).

The organization of this work is as follows. Section introduces the assumptions, data collection, data analysis, and waveform playback. Section presents the experiments and their results. Section reviews these results. Finally, Section concludes.

## Anticogging Proposed Approach

If the torque ripple for a given state of the motor is known, a controller can suppress the ripple simply by commanding a torque that subtracts the ripple torque from the desired torque. Cogging torque is a function of position, so a map of cogging versus position must first be generated. The large number of torque sources, combined with various non-linearities, make the torque ripple map generation challenging. Generating this waveform map is the crux of torque ripple suppression and can be estimated from a number of sources, including commanded position error and accelerations. These values must be measured or converted to units that are useful to the motor driver, typically voltage or current versus position.

Many variables are used throughout this paper and their details are discussed at their introduction.  $\theta$  indicates an angle,  $V$  is a voltage,  $I$  is a current,  $\tau$  is a torque,  $d$  is a duty cycle, and  $f$  is a frequency. These variables can have one or more subscripts.  $m$  indicates a mechanical value,  $e$  is an electrical value,  $i$  and  $j$  are encoder indices,  $clk$  is the microcontroller clock,  $sup$  is supplied,  $des$  is desired,  $app$  is applied,  $cmd$  is commanded,  $est$  is estimated,  $act$  is actual (measured),  $pwm$  is from the pulse width modulation,



**Figure 2.** Diagram of a single half H-bridge inverter connected to one of three phases of a sectioned motor.  $d$  is the high time duty cycle,  $d_{sw}$  is the time it takes for the FETs to switch, and  $f_{pwm}$  is the PWM frequency.

$RMS$  is the root mean squared value,  $min$  is the minimum value,  $max$  is the maximum value,  $pp$  is peak to peak,  $anti$  is anticogged,  $nom$  is nominal (without anticogging),  $fw$  is forward, and  $bw$  is backward. Torque sources are also subscripted.  $cog$  stands for cogging,  $st$  is stiction,  $res$  is resolution,  $frq$  is frequency,  $enc$  is the encoder,  $fr$  is friction, and  $mtl$  is mutual.

## Assumptions

This paper makes the following assumptions which are generally true even for hobby grade motors and ESCs under normal operating conditions:

1. Each motor winding has equal resistance and inductance.
2. A half-H bridge inverter is used to control each phase.
3. The supply voltage and the inverter's current rating are high enough that the motor inductance does not prevent the creation of the desired waveform.
4. Cogging and friction torque ripple are time-invariant.

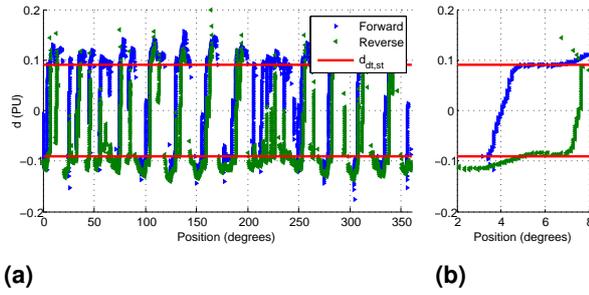
All math for current and voltage is done in signed scalar values, as if the motor is brushed and the supply has positive, negative, and ground rails. Negative values are treated as positive values with  $180^\circ$  added to the electrical position,  $\theta_e$ . The conversion between electrical position and mechanical position,  $\theta_m$ , is  $\theta_e = p\theta_m \bmod 2\pi$  where  $p$  is the number of magnetic pole pairs, as visualized in Figure 2. Control values need to be converted from the desired input quadrature current to phase currents and all feedback needs to be converted from phase currents back to quadrature currents using Clarke and Park transforms. Using these conversions, the motor model can be represented by Eqn. 1.

$$V_{app} = \dot{\theta}_m K_e + IR + L \frac{dI}{dt} \quad (1)$$

where  $V_{app}$  is the voltage applied to the motor,  $K_e$  is the electromotive force constant,  $I$  is the current,  $R$  is the motor resistance, and  $L$  is the motor inductance.

## Waveform Collection

Two methods of collecting the torque ripple waveform were explored. Both exploit the fact that cogging torque is visible



**Figure 3.** Position method collected data showing duty cycle required to hold position from motor M4 in Table 1. This process is described in Section . (a) A full 360° dataset with forward and backward trials and (b) a magnified section showing difference between forward and reverse.

from the mechanical state, i.e. position and speed of the rotor. Both methods are shown in real time in Extension 1.

**Position Based** The position based collection of the current or voltage waveform maps the current or voltage required to maintain a given rotor position. This is done according to Algorithm 1 and is outlined below. An ideal waveform is initially assumed, i.e. trapezoidal for a BLDCM or sinusoidal for a BLACM. A proportional position controller with a high gain commands the rotor to positions according to Eqn. 2 with encoder positions,  $i$ , in monotonically increasing order.

$$\theta_{m,cmd,i} \quad \forall i \in \mathbb{N} \mid \theta_{m,min} \leq \theta_{m,cmd,i} \leq \theta_{m,max} \quad (2)$$

For a motor with continuous rotation the encoder position wraps so the minimum encoder position,  $\theta_{m,min}$ , equals  $\theta_{m,max}$ , the maximum encoder position, and  $i$  spans the full encoder count range. At each commanded position  $i$ , measurements are recorded including: the actual position,  $\theta_{m,act,i}$ , applied PWM duty cycle in Per Unit (PU or %/100),  $d_i$ , supply voltage,  $V_{sup,i}$ , and current,  $I_i$ .

Upon each new command, the motor must come to a complete stop and  $dI/dt = 0$  before sampling data so that Equation 1 can be simplified to  $V_{app} = IR$ . Since the motors do not always go to commanded positions, inconsistencies can occur where

$$\theta_{m,act,i} = \theta_{m,act,j} \quad \forall i \neq j$$

In these cases, the lower magnitude values are discarded.

The above process is repeated commanding  $\theta_{m,cmd,i}$  with  $i$  monotonically decreasing to find the waveform map in the reverse direction. Figure 3 displays these waveforms taken from the experiments outlined in Section . Note that rotating in the reverse direction results in significantly different mapping.

**Acceleration Based** Algorithm 2 is used to map the rotor velocities versus rotor positions under a constant duty cycle. We can then determine the current or voltage waveform based on the rotor accelerations by differentiating the velocities. As in the position based method, an ideal waveform is initially assumed. The motor begins at rest. The PWM duty cycle is incremented for each time step that the motor is stationary. The lowest duty cycle that starts the motor and allows continuous rotation is  $d_{max}$  (not to be confused with the maximum possible  $d = 1$ ),

---

#### Algorithm 1 Position Based Waveform Collection

---

```

for all  $i$  such that  $\theta_{m,min} \leq \theta_{m,cmd,i} \leq \theta_{m,max}$  do
  Command  $\theta_{m,cmd,i}$ 
  while  $\dot{\theta}_m \neq 0$  do
    Wait
  end while
   $\theta_{m,act,i} \leftarrow \theta_{m,act}$ 
   $d_i \leftarrow d$ 
   $V_{sup,i} \leftarrow V_{sup}$ 
   $I_i \leftarrow I$ 
end for

```

---



---

#### Algorithm 2 Acceleration Based Waveform Collection

---

```

 $d_{max} \leftarrow 0$ 
while  $!(\dot{\theta}_m \neq 0 \quad \forall \theta_{m,i})$  do
  if  $\dot{\theta}_m = 0$  then
     $d_{max} \leftarrow d_{max} + \min \Delta d$ 
  end if
  Command  $d_{max}$ 
end while
 $d_{min} \leftarrow d_{max} - \min \Delta d$ 
while  $\dot{\theta}_m \neq 0$  do
  Wait one revolution
   $d_{min} \leftarrow d_{min} - \min \Delta d$ 
  Command  $d_{min}$ 
end while
 $d_{min} \leftarrow d_{min} + \min \Delta d$ 
Command  $d_{max}$ 
Wait  $\bar{\theta} = \text{steady state}$ 
Command  $d_{min}$ 
Wait  $\bar{\theta} = \text{steady state}$ 
 $j \leftarrow 0$ 
while Rotations  $< n$  do
   $\theta_{m,j} \leftarrow \theta_m$ 
   $\dot{\theta}_{m,j} \leftarrow \dot{\theta}_m$ 
   $t_j \leftarrow t$ 
   $j = j + 1$ 
end while

```

---

and is the lowest duty cycle that overcomes the largest cog, stiction, and deadtime (the period of time in switching when no current flows, detailed in Section ). The duty cycle is decremented until the motor stops, then incremented once to find the duty cycle,  $d_{min}$ , that runs the motor at the minimum open loop speed. The motor is restarted by commanding  $d_{max}$  until it reaches a steady-state average speed, then  $d_{min}$  is commanded. The test period is long enough to capture the majority of encoder locations  $m$ , storing position,  $\theta_m$ , and its time derivative,  $\dot{\theta}_m$ .  $\dot{\theta}_m$  is sampled by counting encoder counts in a set time period or counting the time period to see a set number of encoder counts. Repeating this process in the opposite direction yields cogging waveforms similar to the original direction (unlike the Position Based method).

#### Waveform Analysis

For cogging compensation, the data collected in Algorithms 1 and 2 must be converted to voltage or current waveforms,  $I_{cog,i}$  or  $V_{cog,i}$ . It is not guaranteed that a  $V_{sup,i}$  exists for

**Table 1.** Motors and Results of Anticogging with 300 count PWM at 5 volt. M1 is a Maxon EC 45 251601. M2 is an E-flite Park 400 EFLM1300. M3 is an E-flite Park 300 EFLM1150. M4 is an Exceed RC Rocket 86MA10. M5 is a Turnigy Sk3542. M6 is an ElectriFly Rimfire GPMG4555.

Value	Unit	M1	M2	M3	M4	M5	M6
R	$m\Omega$	1030	260	330	220	48	400
$K_v$	rpm/V	285	740	1380	710	1000	750
$I_0$	mA	183	550	400	600	1700	600
$V_0$	V	24	10	8	10	8.4	10
P	Poles	16	14	14	14	14	14
Mass	g	110	56	24	60	142	54
$\tau_{max}$	Nmm	77.7	90.3	48.4	134	363	127
Cost	USD	132	45	38	14	28	55
Dia	mm	42.8	28	28	28	38	28
Length	mm	21.3	29	22	30	44	30
$\tau_{pp}$ nom	Nmm	3.6	5.5	4.2	16.0	38.4	8.7
$\tau_{pp}$ pos	Nmm	1.7	2.8	2.4	9.2	19.7	4.2
$\tau_{pp}$ acc	Nmm	1.7	3.9	1.7	5.4	12.0	3.0
Reduction	%	53	49	60	66	69	65
$\tau_{res}$	Nmm	0.54	0.83	.31	1.02	3.3	0.53
$d_{dt}$	PU	.072	.082	.083	.082	.080	.090
$V_{st}$	mV	0	26.6	16.7	42.1	89.6	117
$\tau_{RMS}$ nom	Nmm	0.57	1.26	0.44	2.30	-	-
$\tau_{RMS}$ anti	Nmm	0.10	0.13	0.07	0.60	-	-
Reduction	%	78	81	88	73	-	-
TRR nom	PU	0.035	0.090	0.051	0.110	-	-
TRR anti	PU	0.008	0.011	0.010	0.029	-	-
$f_{pwm}$ meas	hz	11000	6100	14000	4600	-	-
$f_{pwm}$ est	hz	8100	4600	8100	6100	-	-

all  $i$  from the position method, nor a  $\dot{\theta}_{m,j}$  for all  $j$  in the acceleration method. Fast Fourier Transforms (FFTs) and bi-cubic splines have been used for fitting similar voltage, current, or velocity waveforms in order to fill gaps in collected data and make the data differentiable Moreira (1992). FFTs are of particular interest since most ripple sources are periodic with respect to the mechanical angle Le-Huy et al. (1986). Unfortunately, the raw data cannot be directly fit. Two values, deadtime (explained below) and static friction (also called stiction), complicate matters.

Inverters used to generate waveforms can take one of four states at any given time: high-side transistor conducting, low-side transistor conducting, both conducting, and neither conducting. It is undesirable for both to be conducting, as the inverter will have shoot-through current damaging the circuit. Supply level voltages are produced when only high or only low are conducting, and utilizing PWM between the two an intermediate voltage can be approximated. When neither conduct, the voltage floats or current is sent through flyback diodes. This state is used in 120° commutation on one phase at all times. *Deadtime*,  $d_{dt}$ , is known as the short period when neither conduct while switching between low and high and vice versa so that it can be guaranteed that both transistors never conduct at the same time<sup>†</sup>. For accurate open-loop voltage control (via PWM) the controller must account for this deadtime so that the transistors have the desired on-time pulse ratio. This can be accomplished by adding  $d_{dt}$  (in PU) to the commanded on-time PWM pulse,  $d$  (in PU). The

effective applied voltage due to deadtime is:

$$V_{app} = \begin{cases} V_{sup}(d - d_{dt}) & \text{if } d - d_{dt} \geq 0, \\ 0 & \text{if } d - d_{dt} < 0. \end{cases} \quad (3)$$

where  $V_{sup}$  is the DC supply voltage.

If the deadtime is not already known and compensated for by the driver, the data collected using the position based method, Algorithm 1, is sufficient to determine  $d_{dt}$  using Algorithm 3. All measured duplicates of  $\theta_{m,act}$  are consolidated by storing the maximum and minimum commanded duties and currents in  $d_{max,i}$ ,  $d_{min,i}$ ,  $I_{max,i}$ , and  $I_{min,i}$  respectively. The averages of these are the cogging waveforms,  $d_{cog,i}$  and  $I_{cog,i}$ . Half of the maximum difference of the duty cycle across the motor's position range is the duty cycle required to overcome the maximum deadtime and stiction, denoted  $d_{dt,st,max}$ . All commanded duty cycles with magnitudes below  $d_{dt,st,max}$  correspond to overcoming both stiction and deadtime and are averaged to get  $d_{dt,st}$ . All commanded duty cycles with magnitudes above  $d_{dt,st}$  correspond to overcoming stiction only. The mean of these duty cycles,  $\bar{d}_{st,k}$ , is subtracted from  $d_{dt,st}$  to find the deadtime duty cycle,  $d_{dt}$ . Likewise, the stiction current, denoted  $I_{st}$ , is the mean of half of the current range at each position.

Stiction manifests as a torque. In the open loop case it can be compensated for with a voltage,  $V_{st}$ , since at steady

<sup>†</sup> Deadtime refers to only the time that neither transistor is conducting, and not deadzone, the range of mechanical position slop.

currents and no velocity voltage is linear with current,  $I_{st}$ , and thus is linear with torque. However, because deadtime is a time, it is compensated by modifying the PWM duty cycle on-time by  $d_{dt}$ , in both current and voltage control.

The effects of deadtime and stiction are shown in Figure 3b. The average  $\pm d_{dt,st}$  is shown as horizontal lines. Note that the duty cycles between those lines do not produce motion.

Once deadtime and stiction have been identified, the voltage or current waveforms can be extracted. When using the position method,  $I_{cog,i}$  falls out from Algorithm 3 and  $V_{cog,i}$  can be found using  $d_{cog,i}$  as  $d$  in Equation 3.

When using the acceleration method, the accelerations are found by taking the time derivative of the FFT fitted speeds,  $\ddot{\theta}_{m,i} = \frac{d\mathcal{F}\dot{\theta}_{m,j}}{dt_j}$ . Noting that the rotor inertia,  $J$ , is constant, the cogging torque is:

$$\tau_{cog,i} = J\ddot{\theta}_{m,i} \quad (4)$$

The motor parameters can be used to find the mapping between  $\tau_{cog,i}$ ,  $I_{cog,i}$ , and  $V_{cog,i}$ . If  $J$  is not given,  $d_{min}$  with Equation 3 can be used to scale the acceleration waveform to find  $V_{cog,i}$ , then to  $\tau_{cog,i}$  and  $I_{cog,i}$ .

### Waveform Suppression

For either current or voltage control, FFTs are fitted to the data with respect to mechanical position as mentioned in Section . The fits can be evaluated on the controller in runtime for low orders. Alternatively, a lookup table indexed by encoder position  $i$ , similar to Equation 2, stores precomputed fitted values for  $V_{cog,i}$  or  $I_{cog,i}$ . Stiction values could also be position dependent, but require more analysis to compute than in Algorithm 3. These values are added to the desired voltage or current,  $V_{des}$  or  $I_{des}$  as indicated in the following:

$$V_{out} = V_{des} + \text{sgn}(V_{des})V_{st,i} + V_{cog,i} \quad (5)$$

$$d = \frac{V_{out}}{V_{sup}} + \text{sgn}(V_{out})d_{dt} \quad (6)$$

or

$$I = I_{des} + \text{sgn}(I_{des})I_{st,i} + I_{cog,i} \quad (7)$$

The suppression of cogging torque involves varying current,  $I$ , which adds additional mutual and reluctance torque ripples. With the assumption that mutual and reluctance torques are linear with current, and noting that the feedback throughout this process,  $\theta_m$ , is a mechanical value and thus captures all torque ripple sources, these additional torques are already compensated for within the algorithm.

### Ripple Modeling

In our previous work Piccoli and Yim (2014), we found that the anticogging performance changed with the PWM resolution. To understand this, we model the sources of torque ripple to determine the goodness of the anticogging wave fit as well as evaluate design parameter tradeoffs, chiefly PWM resolution. Outlined below are six identified sources that combine to form our model: PWM resolution ( $\tau_{res}$ ), PWM frequency ( $\tau_{frq}$ ), deadtime ( $\tau_{dt}$ ), encoder phase shifting ( $\tau_{enc}$ ), cogging torque ( $\tau_{cog}$ ), friction torque ( $\tau_{fr}$ ),

---

### Algorithm 3 Position Based Waveform Analysis

---

```

for all  $i$  such that  $\theta_{m,min} \leq \theta_{m,cmd,i} \leq \theta_{m,max}$  do
  for all  $j$  in range of  $\theta_{m,act,j}$  do
    if  $\theta_{m,cmd,i} = \theta_{m,act,j}$  then
      if  $d_j > d_{max,i}$  then
         $d_{max,i} \leftarrow d_j$ 
         $I_{max,i} \leftarrow I_j$ 
      end if
      if  $d_j < d_{min,i}$  then
         $d_{min,i} \leftarrow d_j$ 
         $I_{min,i} \leftarrow I_j$ 
      end if
    end if
  end for
   $d_{dt,st,i} \leftarrow \frac{d_{max,i} - d_{min,i}}{2}$ 
   $d_{cog,i} \leftarrow \frac{d_{max,i} + d_{min,i}}{2}$ 
   $I_{st,i} \leftarrow \frac{I_{max,i} - I_{min,i}}{2}$ 
   $I_{cog,i} \leftarrow \frac{I_{max,i} + I_{min,i}}{2}$ 
end for
 $d_{dt,st,max} = \max_i d_{dt,st,i}$ 
 $k \leftarrow 0$ 
for all  $i$  such that  $d_{dt,st,max} > d_{max,i} \mid -d_{dt,st,max} < d_{min,i}$  do
   $d_{dt,st,ktemp} \leftarrow d_{dt,st,i}$ 
   $k \leftarrow k + 1$ 
end for
 $\bar{d}_{dt,st} = \bar{d}_{dt,st,ktemp}$ 
 $k \leftarrow 0$ 
for all  $i$  such that  $d_{dt,st,max} < d_{min,i} \mid -d_{dt,st,max} > d_{max,i}$  do
   $d_{st,k} \leftarrow d_{dt,st,i}$ 
   $V_{st,k} \leftarrow d_{dt,st,i} V_{sup,i}$ 
   $k \leftarrow k + 1$ 
end for
 $V_{st} \leftarrow \bar{d}_{st,i} \bar{V}_{sup,i}$ 
 $I_{st} \leftarrow \bar{I}_{st,i}$ 
 $d_{dt} \leftarrow d_{dt,st} - \bar{d}_{st,i}$ 

```

---



---

### Algorithm 4 Acceleration Based Waveform Analysis

---

```

 $\ddot{\theta}_{m,i} = \frac{d\mathcal{F}\dot{\theta}_{m,j}}{dt_j}$ 
for all  $i$  such that  $0 \leq \theta_{m,i} \leq 2\pi$  do
   $d_{cog,i} \leftarrow d_{min} \ddot{\theta}_{m,i} / \max_i \ddot{\theta}_{m,i}$ 
end for

```

---

and mutual torque ripple ( $\tau_{mtl}$ ). All torque ripple sources are modeled as RMS and assumed to have no covariance so that the total RMS torque ripple can be calculated with:

$$\tau_{RMS} = \sqrt{\tau_{res}^2 + \tau_{frq}^2 + \tau_{dt}^2 + \tau_{enc}^2 + \tau_{cog}^2 + \tau_{fr}^2 + \tau_{mtl}^2} \quad (8)$$

**PWM resolution** torque ripple,  $\tau_{res}$ , stems from the discretization of the desired waveform, where the desired waveform is both the standard sine or trapezoidal signal as well as its change in amplitude according the anticogging waveform, so both nominal motor control as well as anticogged motor control are affected. The error is approximately a sawtooth wave, thus the RMS ripple is the

amplitude over  $\sqrt{3}$ , which is:

$$\tau_{res} = V_{sup} f_{pwm} K_{\tau} / (R f_{clk} \sqrt{3}) \quad (9)$$

where  $f_{pwm}$  is the PWM frequency,  $K_{\tau} = 60 / (2\pi K_v)$  is the torque constant, and  $f_{clk}$  is the clock frequency. In practice, this is a lower bound since the error is not an exact sawtooth.

**PWM frequency** is a design parameter so we should model its effect on torque ripple. During PWM, when the pulse is on, the driver drives current through the coils in the motor to cause a torque on the rotor, and not when the pulse is off. It is best that this is faster than the motor's time constant,  $\tau_{pwm} = L/R$ , since a notable torque ripple is produced at the PWM frequency if it is slower. The effective frequency is that of the duty cycle's on time,  $\omega_{pwm} = 2\pi f_{pwm} / d$ . Since high rates are not guaranteed, the AC RMS of a low passed signal, like  $\tau_{frq}$ , is:

$$\tau_{frq} = \frac{V_{sup} K_{\tau} \sqrt{d} \sqrt{1-d}}{R \sqrt{1 + \tau_{pwm}^2 \omega_{pwm}^2}} \quad (10)$$

**Deadtime torque ripple** is the ripple caused by the cessation of current flowing through the motor during the transistor switching time. As the frequency of the PWM increases, the switching deadtime,  $d_{dt}$ , becomes a larger portion of the PWM period. We assume the PWM frequency is above  $1/(2\pi\tau)$  so that the PWM voltage is low pass filtered to be  $V_{sup}d$ . If  $d_{dt}$  is large at frequencies below  $1/(2\pi\tau)$ , the inverter used is too slow for the motor and is likely too large for the motor. With this assumption, the deadtime torque ripple,  $\tau_{dt}$ , is:

$$\tau_{dt} = \frac{V_{sup} d K_{\tau} \sqrt{d_{dt}} \sqrt{1-d_{dt}}}{R} \quad (11)$$

**Encoder phase shift** is another source of torque ripple that comes from delays in sensing and calculation. If the controller makes PWM updates at the PWM frequency and the PWM frequency is lower than the encoder's change in position rate, then the controller misses position steps. In reality, this always happens to some extent, even with high PWM frequencies, unless updates are interrupt driven. This encoder phase shift affects both anticogging and the nominal commutation, so both waveforms must be known. This value is velocity dependent and should be calculated for the motor's intended velocity. A Monte Carlo simulator simulates calculation start times and various sample times between calculations to find the distribution of encoder phase shifts. Because the encoder is discrete, the simulator returns a vector,  $\vec{t}_{enc}$ , of length  $1 + \lceil f_{enc}/f_{pwm} \rceil$  containing relative times spent with phase shifts of 0 to  $\lceil f_{enc}/f_{pwm} \rceil$  encoder counts. The RMS of each phase shift is calculated and stored in  $\vec{V}_{\Delta i}$ . These are combined by taking the square root of the sum of the squares of the element wise product or Hadamard product,  $\odot$ , of the times and voltages,  $V_{enc} = \sqrt{\sum (\vec{V}_{\Delta i} \odot \vec{t}_{\Delta i}) \odot (\vec{V}_{\Delta i} \odot \vec{t}_{\Delta i})}$ , then converted to a torque with  $\tau_{enc} = V_{enc} K_{\tau} / R$ .

**Cogging torque** is the primary concern of this paper. We set  $\tau_{cog} = \text{RMS}(\tau_{cog,i})$ , which is the RMS of the data sampled with algorithm 1 and analyzed with algorithm 3.

**Friction torque** ripple is from position dependent friction, perhaps from eccentricities in the motor's bearings. To

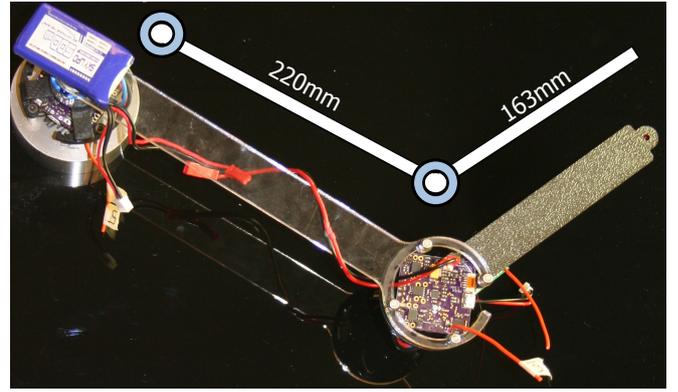


Figure 4. Top view of the robotic arm.

find friction's effect, we perform algorithm 1 in both the forward and backward direction, giving us  $\tau_{cog,i,fw}$  and  $\tau_{cog,i,bw}$ . The friction torque ripple is then found with  $\tau_{fr} = \text{RMS}((\tau_{cog,i,fw} - \tau_{cog,i,bw})/2)$ .

**Mutual torque ripple** is from a mismatching of inverter to back EMF waveforms. While it is possible to apply any one of the many mutual torque correction algorithms outlined in section , for simplicity we assume this is not done. This ripple is typical of driving an ideal waveform (trapezoidal or sinusoidal) on a motor with a non-ideal back EMF shape. If the back EMF shape does not match the driven voltage shape, the error between the waves grows linearly with voltage amplitude and is zero at no voltage amplitude. This ripple could be modeled by sampling the back EMF while the motor is generating, then comparing the sampled wave to the driven wave. Since in our tests we do not put the motor under any load, which is proportional to current, other than cogging and friction, any mutual torque ripple will be minimal and we ignore it in our model.

## Design and Experimental Results

To demonstrate the applicability of the proposed technique for robot arms, a two degree of freedom (DOF) planar robot arm was created that displays smooth motion suitable for simple tool-mediated haptic rendering. A model for specifications of this arm is the popular commercial haptic device, the PHANTOM Omni, now called the Geomagic Touch Geomagic (2015). The arm specifications includes a planar 2 DOF subset of the Geomatic Touch workspace. This workspace is advertised as rectangular area ( $160 \times 120$ ) mm. However, it is a polar device with workspace measurements  $100 \text{ mm} < \text{radius} < 270 \text{ mm}$  and  $90^\circ$  in angular range. The maximum continuous force output is 880 mN.

Figure 4 shows the two link serial chain used for final validation of the presented method. The length of the first link between the motors is 220 mm and the second link between the second motor and the end effector is 163 mm. This has an effective envelope of  $57 \text{ mm} < \text{radius} < 383 \text{ mm}$ , and  $360^\circ$  angular range which encompasses the required ( $160 \times 120$ ) mm workspace. An onboard wireless radio combined with a battery powering the second joint allows control of the second joint without wires crossing the first joint, reducing external friction sources. Encoders with 12 bit resolution (4096 count) yield a translation positional

resolution of 0.087 mm when the links are parallel and 0.59 mm when the links are perpendicular. This is large compared to the 0.06 mm resolution of the Geomagic Touch, but is one sacrifice for obtaining a low cost yet larger workspace.

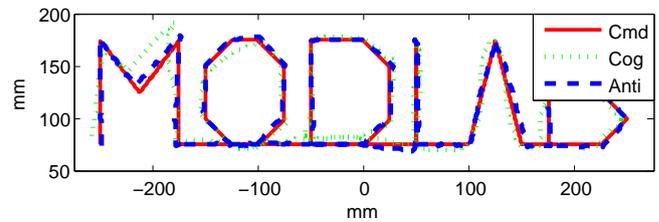
With the arm lengths chosen, the motor torque required to generate desired max force can be determined. The translation forces applicable by the end effector depend on the joint angles. The nominal position is defined to be identical to the Geomagic Touch with the second joint at  $90^\circ$ . The maximum applicable force occurs with the shorter lever arm creating the largest static force. This gives a target max motor torque of  $0.88 \text{ N} \cdot 163 \text{ mm} = 143 \text{ N mm}$ .

### Experimental Setup

To determine the most suitable motor, various motors of the appropriate size were evaluated before and after anticogging was applied, but without robot arm links attached to ensure the only sensed torque was from cogging torque. Experiments used a custom motor controller and driver. A Texas Instruments TMS320F28035 provides indirect field-oriented control at 100 kHz. A 600 W, 3 phase inverter, pulse-width modulated at 50 kHz symmetrically (up/down), enables updates at 100 kHz with a 300 count PWM. A diametrically aligned magnet affixed to the rotor of each motor and an Austria Microsystems AS145B 12-bit (4096 count) magnetic rotary encoder attached to the stator measure position. The cost of this encoder and magnet pair is \$6.69 USD at quantity of 1000 with similar solutions as low as \$1.91 USD using the AS5601. The encoder magnet pair is the only required addition to standard hobby ESCs.

The final version of the arm uses this setup with identical hardware, except an updated motor driver using an STMicroelectronics STM32F373 controls phase voltages at 10 kHz with a 1000 count PWM resolution. Haptic feedback is difficult to show in a visual form, so trajectory following was chosen instead to demonstrate smoothness. An example 2.035 m trajectory consisting of 36 line segments, as seen in Figure 5 and Extension 1, represents a simplified example of a path. Commands are sent and feedback is received synchronously at 150 Hz. Encoder positions are used to calculate the end effector position. We further explore the trade-offs between high PWM frequency versus high PWM resolution (low PWM frequency) with this motor driver, where frequency and resolution are related by  $V_{res} = \lfloor 72000000 / f_{pwm} \rfloor V_{sup}$ .

For validating the proposed acceleration and position waveform generation methods and measuring frequency-resolution trade-offs, a third method is used to provide ground truth, experimentally determining torque ripple. It uses an external torque sensor, an ATI Industrial Automation Nano17 six-axis force and torque transducer with 1/64 N mm resolution, sampled while performing the acceleration method for five seconds at 20 kHz in MATLAB. We apply a notch filter at the motor and torque sensor's natural frequency to eliminate noise generated by the test apparatus. These frequencies range from 300 Hz to 500 Hz, which is too low to interfere with ripple generated by the PWM resolution, PWM frequency, or encoder shift and too high to be from cogging or mutual torques when spinning at the low test speeds of roughly 1 Hz. The motor's datasheet



**Figure 5.** Trajectory of robotic arm with and without anticogging. Cmd is the commanded trajectory, Cog is the actual trajectory without compensation, and Anti is the actual trajectory with anticogging enabled.

provides motor constants to translate torque to voltage and current for this third method. Values of  $\theta$ ,  $\dot{\theta}$ ,  $V_{sup}$ , and  $d$  are read at 1 kHz from the controller.

The original controller and driver are tested with eleven motors, demonstrating anticogging's efficacy across a wide range of motors. Six motors are used throughout this paper as examples and are indicated in Table 1. We perform additional tests on motors M1, M2, M3, and M4 in search for an optimal frequency versus resolution trade-off. We use the measured RMS torques from the torque sensor to validate the estimated torque ripple sources from section for each tested PWM frequency.

### Results

A common metric of torque ripple is the torque ripple factor (TRF) Le-Huy et al. (1986) Qian et al. (2004). The equation for TRF is:

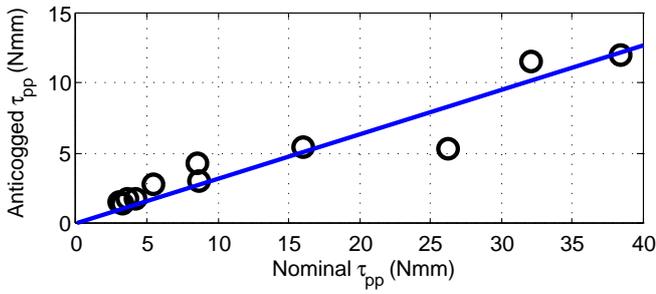
$$TRF = \frac{\tau_{pp}}{\bar{\tau}} \quad (12)$$

where  $\tau_{pp}$  is the peak to peak torque variation and  $\bar{\tau}$  is the average applied torque. For mutual and reluctance torque ripple this measurement is constant over different commanded torques, as both torque ripple and desired torque are linear with current and thus  $\bar{\tau}$ . Since cogging torque is independent of current and thus  $\bar{\tau}$ , TRF is not constant and is less useful. TRF is infinite for all motors at zero applied torque because there is still torque ripple from cogging. In place of  $\bar{\tau}$ , a divisor that remains constant for each motor is proposed as *Torque Ripple Ratio* or *TRR*, defined as follows:

$$TRR = \frac{\tau_{pp}}{\tau_{max}} \quad (13)$$

where  $\tau_{max}$  is the maximum continuous torque that the motor can apply, which can be derived from the motor's datasheet by multiplying the maximum continuous current and the torque constant. Using this metric, Figure 1 shows the relationship between torque ripple and price of 11 arbitrary BLDCMs; a notable inverse correlation before anticogging is evident, while after anticogging the metric is relatively constant.

Figure 6 shows a plot of the before and after results of applying anticogging to the 11 tested motors with a 300 count PWM resolution. A line fit shows a 69% average reduction in torque ripple. Table 1 shows the details for a subset of the motors from Figure 6.  $\tau_{pp \text{ nom}}$  is the nominal peak to peak cogging torque of the motor.  $\tau_{pp \text{ pos}}$  and  $\tau_{pp \text{ acc}}$  are the peak to peak cogging torques after applying the position method and the acceleration method, respectively.



**Figure 6.** Torque ripple after anticogging versus torque ripple before anticogging for eleven tested motors. Fit line is  $y = 0.3139x$  with an  $R^2 = 0.8922$ .

Data from the PWM frequency versus resolution tests are also in table 1.  $\tau_{RMS\ nom}$  is the RMS torque ripple without anticogging, while  $\tau_{RMS\ anti}$  is the RMS torque ripple with anticogging. Reduction shows the percent torque ripple reduction of the motors. All motors have an RMS reduction of greater than 70% and peak at 88% when using the appropriate PWM frequency and is shown by comparing  $\tau_{nom,act}$  in Figures 7 and  $\tau_{anti,act}$  in 8. In contrast to the fixed resolution reductions, picking the proper PWM resolution removes up to  $3\frac{1}{3}$  times the peak-to-peak ripple.

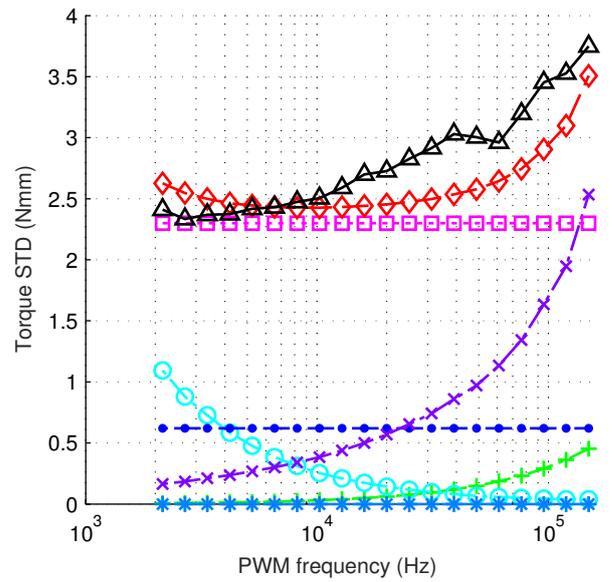
The values  $f_{pwm\ meas}$  and  $f_{pwm\ est}$  compare the measured and estimated ideal frequencies. Frequency tests are done at the frequencies of  $1100 \times 1.33^x$  where  $x = 0-17$  inclusive, giving eighteen frequencies spanning 1100 hz–140 khz. We calculate the estimated RMS torque ripples at the same frequencies and take one with the minimum RMS torque as the best. The estimated best frequencies are all within two calculated frequencies of their best measured, and three out of four were within one. Figure 7 and Figure 8 show the components of the RMS torques for motor M4 before anticogging and after anticogging respectively. Figure 9 shows the four motors' RMS torques versus PWM frequencies. The data plateaus near the minima, particularly in motors M1 and M3, explaining the small discrepancies in frequency.

A metric that describes the value of a motor is  $\tau_{pp} \times cost$ . From the results in Table 1, motor M2 has the best value before compensation, but motor M4, a motor that fills the same niche in terms of size, torque, and power, wins out after compensation. Conveniently, M4 is also the least expensive of the tested motors.

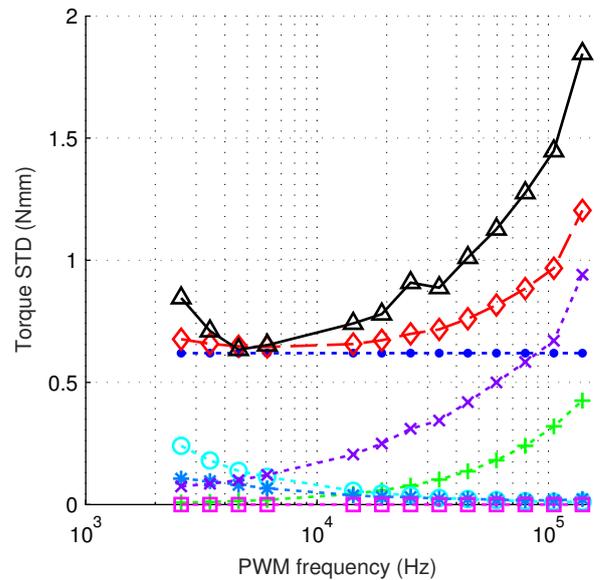
Since it has the best value, we use motor M4 on the haptic arm, noting it has the highest TRR of the tested motors. The effect of cogging on the end point position is clearly evident in Figure 5 and Extension 1. The cartesian root mean squared position error (RMSE) with cogging compensation turned off is 7.38 mm, while the RMSE with acceleration type anticogging on is 3.52 mm.

## Discussion

The data presented in Figure 1 shows that anticogging gives a low cost motor a TRR lower than that of a motor that is nearly an order of magnitude more expensive. Even with a low resolution of 300 PWM counts across 5 volt, there is an average  $\tau_{pp}$  reduction of 69%. At higher resolutions as much as an 88.2% reduction has been seen. Using anticogging,



**Figure 7.** Motor M4 RMS torque versus PWM frequency with anticogging disabled. (+) is  $\tau_{res}$ , (o) is  $\tau_{frq}$ , (\*) is  $\tau_{cog}$ , (.) is  $\tau_{fr}$ , (x) is  $\tau_{dt}$ , (□) is  $\tau_{cog}$ , (◇) is  $\tau_{nom,est}$ , (△) is  $\tau_{nom,act}$

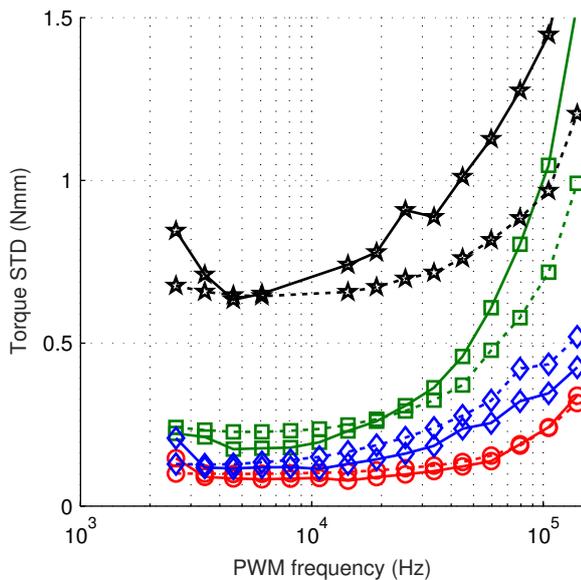


**Figure 8.** Motor M4 RMS torque versus PWM frequency with anticogging enabled. (+) is  $\tau_{res}$ , (o) is  $\tau_{frq}$ , (\*) is  $\tau_{cog}$ , (.) is  $\tau_{fr}$ , (x) is  $\tau_{dt}$ , (□) is  $\tau_{cog}$ , (◇) is  $\tau_{anti,est}$ , (△) is  $\tau_{anti,act}$

the cartesian RMS position error of a direct drive arm's end effector can be reduced to less than half.

## Comparison of Methods

Verifying that both methods of cogging characterization map the torque ripple accurately is crucial. Figure 10 displays both methods as well as the ground truth from the external torque sensor detailed in Section . From the plot, the reader can see that all three methods are in agreement in shape, while the position method differs slightly. This is not to say that the position is more or less accurate. Because the speed method and external torque sensor did readings at the same time, they both detect the added mutual torque from bearing

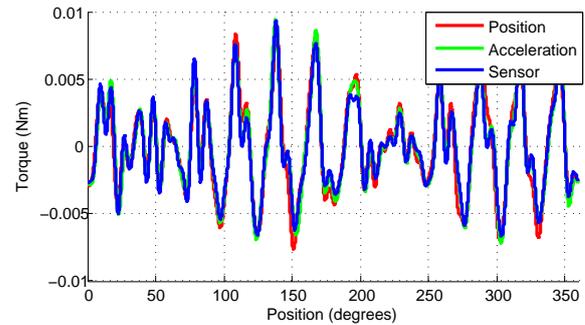


**Figure 9.** Motors M1, M2, M3, and M4 anticogged RMS torque versus PWM frequency predicted and measured. Solid lines are measured and dashed lines are predicted. ( $\circ$ ) is M1, ( $\square$ ) is M2, ( $\diamond$ ) is M3, and ( $\star$ ) is M4.

friction, while the position method does not. All successful characterizations have a RMS torque error of  $< 1$  N mm.

It is mentioned in Section that speed control loops have limited success suppressing torque ripple, yet the acceleration method, which uses speed feedback, maps cogging torque well. One reason is that the cog mapping is done offline at the lowest possible open loop speed, and thus sensor delay has less impact with respect to position. Furthermore, in a control loop, there must be error to correct and the reactions cause further delays. Another factor may be that all motors tested were smaller hobby or robotics motors in the 18 W to 670 W range. Small size yields smaller inertia as indicated by Equation 4, which gives larger, and thus more measurable accelerations for the same torque. The results may not be as favorable for higher inertia motors, motors with higher minimum speeds, or lower frequency speed sensing.

The position method also tracked cogging torque well, despite being based on a different principle. Unlike the acceleration method, which loses DC signal<sup>‡</sup> values when taking the derivative, the position method overcomes both the oscillating cogging torque and DC signal friction. Although constant values are easily characterized and compensated, the characterization does introduce a failure mode. The extracted values for deadtime generally agree across motors, as seen in Table 1. A supplementary test using a current sensor and the torque sensor on motor M4 found that, while current production starts at  $d = 0.071$ , external torque is not felt until  $d = 0.083$ . This indicates that the deadtime  $d_{dt} = 0.071$  is the deadtime duty cycle for this motor driver and the stiction is  $d_{st} = 0.012$  or  $V_{st} = 60$  mV at the tested location. The discrepancy between these values and those in Table 1 could be because stiction is not consistent across the full range of motion, but the calculations for the compensation assume stiction is consistent. The expensive M1 motor has no



**Figure 10.** Fitted cogging torque ripple data sampled via the position method, acceleration method, and torque sensor versus position on an Exceed RC 86MA10 motor. Voltages are converted to torques using motor datasheet parameters where required.

detectable stiction, perhaps contributing to its more accurate estimation of  $d_{dt} = 0.072$ .

In the process of testing, it was found that with low gains on the position controller, deadtime was not visible in the data. As always, proportional gains that are too high cause the controller to go unstable; thus, gains must be chosen wisely. Excessive gains occasionally prevented more than one iteration of anticogging using the position method.

With a sufficient quality cog map loaded into the driver's onboard memory, the fidelity of the output waveform is dependent on the controller speed and resolution. At the maximum tested motor speeds (roughly 100 RPM), the encoder incremented around 7 kHz but the controller's loop speed was significantly higher at 100 kHz, indicating that the loop speed was not a factor. The PWM resolution during these tests were 300 counts across a voltage of 5 V, resulting in 0.017 V increments. Converting this voltage increment into torque increments for each motor using datasheet parameters,  $\tau = K_{\tau}I$ , and  $V = IR$ , gives the values indicated by  $\tau_{res}$  in Table 1. It can be seen that the resolutions are on the same order of magnitude as the anticogged  $\tau_{pp}$ , between 1 and 5 counts across the full range of motors. This indicates that PWM resolution is the limiting factor of torque ripple reduction in this dataset.

This prompted the PWM resolution versus frequency modeling and tests on motors M1, M2, M3, and M4. Figure 7 shows that at high frequencies, where the deadtime is a significant portion of the period,  $\tau_{dt}$  is the driving torque ripple source with the exception of  $\tau_{cog}$ . Figure 8 supports that at low PWM resolutions (high PWM frequencies) the RMS torque from resolution error,  $\tau_{res}$ , also follows the total anticogged RMS torque,  $\tau_{RMS}$ . This, however, is not the whole story. From Figure 8, it can be seen that friction torque is the leading contributor of torque ripple for all of the motors at most frequencies. Thus, friction torque is justly mentioned in Holtz and Springob (1996) as a torque ripple source. Torque ripple from low PWM frequencies show a sharp increase in RMS ripple at the lowest of frequencies. Particularly on the motors with less inertia, motor vibrations

<sup>‡</sup>When referring to DC components or DC signals, the authors are referring to the non-oscillating offset components in the frequency domain, rather than current.

make cog map generation difficult for the position method since extremely low proportional gains must be used to keep the motor stable. At the low speeds of these tests, torque from encoder delay has a negligible effect, but its value should be calculated at the maximum desired motor speed in real applications.

With the frequency search calculations verified with experiments, we can compute optimal frequencies for new motors. We calculate  $\tau_{pwm}$  and  $\tau_{freq}$  straight from datasheet values. The  $\tau_{dt}$  can be calculated from datasheet values and motor driver knowledge, which can be gathered from the position control method. The portion of  $\tau_{enc}$  from the nominal sinusoidal or trapezoidal line voltages can easily be calculated knowing the desired motor speed, the motor driver's clock, and the encoder resolution. Anticogging's effect on  $\tau_{enc}$  is only known after a cog map is generated. Mutual torque's contribution requires a high quality simulator and model or for the motor to be in hand. While it's effect can be measured with a torque sensor or current sensor, it is best to apply one of the many mutual torque ripple compensation methods outlined in section . Since  $\tau_{cog}$  and  $\tau_{fr}$  are assumed to be constant, they do not contribute to the PWM frequency decision.

Of the four tested motors, all of the minimum RMS torque frequencies lie between 4 khz and 14 khz as seen in table 1. The differences lie in which ripple sources dominate at each frequency for each motor. For a wide range of typical PWM frequencies, the model tracks the actual RMS well. The model tends to underestimate at the frequency extremes. There may be an unmodeled torque ripple source that is either frequency, PWM duty cycle, or speed dependent, since these all vary proportionally throughout the tests. Perhaps Coulomb friction plays a larger role than expected, as suggested by Holtz and Springob (1996). Despite its these errors, the model not only allows a robot designer to choose the appropriate motor driver frequency, but also predicts the expected amount of RMS torque ripple across a range of motors and frequencies.

Now that we have shown that the anticogging process can suppress torque ripple to a predicted amount and have found the appropriate PWM frequency, we can compare the potential robotic arm motors. If torque ripple is the primary concern, motor M3 is reduced to the lowest  $\tau_{RMS}$  thanks to its small torque resolution step size, while the next lowest, M1, is 4.58 times as massive, 3.47 times as expensive, while having only 1.61 times as much continuous torque. If value is the primary concern, motor M4 wins since motor M1 is 1.83 times as massive, 1.60 times larger, 9.43 times more expensive, 0.58 times as much continuous torque than M4, and has 1.2 times the TRR when M4 is anticogged. This is why we chose motor M4 for the robotic arm.

## Arm Test Results

The results in the previous section guided the design of the updated motor driver used in the robotic arm and is described at the end of Section . Despite the arm having significantly larger inertial loads, which raises the required output torque and lowers the TRF when compared with bare motor cog testing, RMSE decreased by 52% using anticogging. The results are visualized in Figure 5.

Comparing the resulting motor capabilities to the desired robot arm requirements, the maximum continuous force is close to the Geomagic Touch. The M4 has 134 N mm which compares to our target 143 N mm. While most commercial haptic devices do not list torque ripple, they often specify a back-drive friction, which is an error from the desired force output. The Geomagic Touch lists a back drive friction of 0.26 N. Solving  $V_{st}$  of motor M4, the stiction force at the end of the second joint is 0.016 N. For a second comparison, we can normalize the back drive friction with the max force, which gives an effective  $TRR = 0.30$  for the Geomagic Touch. This is quite large compared to the  $TRR$  of the proposed device at 0.04, however the  $TRR$  is cyclical and back drive is not. Human touch sensitivity is noticeably stronger with frequencies  $>5$  Hz Johansson and Flanagan (2009). Nominal human motions move the arm at  $120^\circ$  in 1 second, that would correspond to approximately 5 Hz as the dominant frequency in Figure 10 over  $120^\circ$ . Faster motions would result in higher frequencies to which humans are much more sensitive.

## Conclusion and Future Work

A two DOF robot arm that has low-cost direct drive motors has shown to be comparable to popular commercial devices. The commercial devices in comparison were six DOF so comparing against a subset of the device is not quite fair. However, the specifications of error (from back-drive), range of motion, and torque capability show that the arm can provide useful forces and motions in haptic applications.

Moreover, the arm was shown to have improved performance by using an anticogging control. The impact of this work is much broader than just haptic robot arms as the methods have been tied more specifically to motor characteristics that would work on any robot arm.

Two methods for mapping the cogging torque waveform have been presented. Both methods only require a position sensor, yet both accurately map the cogging torque when compared to an external force sensor. Reductions in torque ripple from 49% to 88% have been seen across the 11 tested motors.

A model for torque ripple as a function of PWM frequency is presented that closely predicts the optimal PWM frequency (the PWM frequency that yields the lowest  $\tau_{RMS}$ ) and accurately reflects the torque ripple over most frequencies.

In most cases, low cost motors with comparable or better peak torque and comparable or smaller size to high-end motors can be anticogged via the proposed methods such that the torque ripple is reduced below that of the more expensive motors. For example, M4 in Table 1 has slightly better mass and max torque, but has double the uncompensated torque ripple and is 30% of the cost of the M6 motor. Comparing the most expensive motor, M1, in it's normal state versus the least expensive motor, M4, anticogged, the  $\tau_{RMS}$  values are comparable at 0.57 for M1 and 0.60 for M4, while the TRR is 0.035 for M1 and 0.029 for M4, which is notably better for M4. Furthermore, M1 is 9.43 times more expensive than M4.

Since friction torque is the largest remaining portion of torque ripple after anticogging for most motors, future work

will be concentrated on implementing friction compensation. Also, online waveform mapping and adjustment while performing arbitrary tasks is the next logical step for motor properties that vary over time, though it's likely cogging torque ripple and stiction change very slowly. Upon successful torque ripple minimization, Hanselman (1994) noted that the next logical step in motor design is to create motors with higher performing back EMF shapes, since simplicity of commutation is no longer a concern due to improved power electronics and microprocessor computation.

As manufacturing structures and assembly become lower cost, actuators will dominate the cost of robotic systems. This is already the case in low-cost robotic systems. Finding methods to use low-cost motors in high performance robotic applications will enable new markets for robot systems.

### Acknowledgements

We would like to thank Matt Lisle, Kris Li, Patrik Roeller, Rui Zhang, Haofang Yuan, James Paulos, and Chris Thorne for their effort developing hardware and performing tests.

### Funding

The authors would like to acknowledge the support of National Science Foundation [grant number 1138847]; and Willow Garage.

### References

- Farhad Aghili. Adaptive Reshaping of Excitation Currents for Accurate Torque Control of Brushless Motors. *Control Systems Technology, IEEE Transactions on*, 16 (2):356–364, March 2008. ISSN 1063-6536. doi: 10.1109/TCST.2007.908213.
- B. Armstrong. Friction: experimental determination, modeling and compensation. In *Robotics and Automation, 1988. Proceedings., 1988 IEEE International Conference on*, pages 1422–1427 vol.3, Philadelphia, April 1988. doi: 10.1109/ROBOT.1988.12266.
- E. Favre, L. Cardoletti, and M. Jufer. Permanent-magnet synchronous motors: a comprehensive approach to cogging torque suppression. *Industry Applications, IEEE Transactions on*, 29(6):1141–1149, nov/dec 1993. ISSN 0093-9994. doi: 10.1109/28.259725.
- Geomagic. Geomagic Touch (formerly Geomagic Phantom Omni) Specifications, August 2015. [Online; accessed 14-August-2015].
- D.C. Hanselman. Minimum torque ripple, maximum efficiency excitation of brushless permanent magnet motors. *Industrial Electronics, IEEE Transactions on*, 41(3):292–300, jun 1994. ISSN 0278-0046. doi: 10.1109/41.293899.
- V. Hayward and K.E. MacLean. Do it yourself haptics: part i. *Robotics Automation Magazine, IEEE*, 14(4):88–104, December 2007. ISSN 1070-9932. doi: 10.1109/M-RA.2007.907921.
- J. Holtz and L. Springob. Identification and compensation of torque ripple in high-precision permanent magnet motor drives. *Industrial Electronics, IEEE Transactions on*, 43(2):309–320, apr 1996. ISSN 0278-0046. doi: 10.1109/41.491355.
- J.Y. Hung and Z. Ding. Design of currents to reduce torque ripple in brushless permanent magnet motors. *Electric Power Applications, IEE Proceedings B*, 140(4):260–266, Jul 1993. ISSN 0143-7038.
- Roland S. Johansson and J. Randall Flanagan. Coding and use of tactile signals from the fingertips in object manipulation tasks. *Nature Reviews Neuroscience*, 10: 345–359, May 2009.
- Young-Hoon Kim and In-Joong Ha. A learning approach to precision speed control of servomotors and its application to a VCR. *Control Systems Technology, IEEE Transactions on*, 7(4):466–477, July 1999. ISSN 1063-6536. doi: 10.1109/87.772162.
- H. Le-Huy, R. Perret, and R. Feuillet. Minimization of Torque Ripple in Brushless DC Motor Drives. *Industry Applications, IEEE Transactions on*, IA-22(4):748–755, july 1986. ISSN 0093-9994. doi: 10.1109/TIA.1986.4504787.
- G.J. Liu and A.A. Goldenberg. Robust control of robot manipulators incorporating motor dynamics. In *Intelligent Robots and Systems, 1993. Proceedings of the 1993 IEEE/RSJ International Conference on*, volume 1, pages 68–75, Tokyo, 1993. IEEE.
- J.C. Moreira. Torque ripple minimization in switched reluctance motors via bi-cubic spline interpolation. In *Power Electronics Specialists Conference, 1992. PESC '92 Record., 23rd Annual IEEE*, pages 851–856 vol.2, Toledo, jun-3 jul 1992. doi: 10.1109/PESC.1992.254794.
- W.S. Newman and J.J. Patel. Experiments in torque control of the adeptone robot. In *Robotics and Automation, 1991. Proceedings of the 1991 IEEE International Conference on*, pages 1867–1872 vol.2, Sacramento, April 1991. IEEE. doi: 10.1109/ROBOT.1991.131897.
- S.J. Park, H.W. Park, M.H. Lee, and F. Harashima. A new approach for minimum-torque-ripple maximum-efficiency control of BLDC motor. *Industrial Electronics, IEEE Transactions on*, 47(1):109–114, feb 2000. ISSN 0278-0046. doi: 10.1109/41.824132.
- V. Petrovic, R. Ortega, A.M. Stankovic, and G. Tadmor. Design and implementation of an adaptive controller for torque ripple minimization in PM synchronous motors. *Power Electronics, IEEE Transactions on*, 15(5):871–880, sep 2000. ISSN 0885-8993. doi: 10.1109/63.867676.
- Matthew Piccoli and Mark Yim. Cogging Torque Ripple Minimization via Position Based Characterization. In *Proceedings of Robotics: Science and Systems*, Berkeley, July 2014.
- W. Qian, S.K. Panda, and J.X. Xu. Torque ripple minimization in PM synchronous motors using iterative learning control. *Power Electronics, IEEE Transactions on*, 19(2):272–279, march 2004. ISSN 0885-8993. doi: 10.1109/TPEL.2003.820537.
- R.S. Wallace and D.G. Taylor. Low-torque-ripple switched reluctance motors for direct-drive robotics. *Robotics and Automation, IEEE Transactions on*, 7(6):733–742, 1991.

## Appendix: Index to Multimedia Extensions

Archives of IJRR multimedia extensions published prior to 2014 can be found at <http://www.ijrr.org>. After 2014 all videos are available on the IJRR YouTube channel at <http://www.youtube.com/user/ijrrmultimedia>

Extension	Type	Description
1	Video	Shows cogging torque, mapping, and haptic arm experiments