

Cogging Torque Ripple Minimization via Position-Based Characterization

Matthew Piccoli

School of Engineering
and Applied Science

University of Pennsylvania

Philadelphia, Pennsylvania 19104–6391

Email: piccoli@seas.upenn.edu

Mark Yim

School of Engineering
and Applied Science

University of Pennsylvania

Philadelphia, Pennsylvania 19104–6391

Email: yim@seas.upenn.edu

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 inherent 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 system that is low cost using a method for *anticoggling* - the compensation of cogging torque in low cost, high torque motors. 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 anticoggling shows validation; however, the approach is much broader, and can be applied to any precision force application. Test results on eleven different motors show an average removal of 69% of torque ripple with no added cost in robotic servo applications.

I. INTRODUCTION

Robots often employ low cost DC brushed motors when performance (speed, torque, and precision) is not a primary concern. When performance is a concern, permanent magnet synchronous motors (PMSM) such as brushless DC motors (BLDCM) and brushless AC motors (BLACM), often work well. 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 periodic fluctuation in the motor torque as the output shaft rotates. This has been recognized as a problem in a variety of robot applications [1] [2] [3]. In haptic rendering it is especially troublesome [4]. Transmissions, such as gear boxes, add non-linear torque variations that are difficult to model and compensate, making direct-drive favorable.

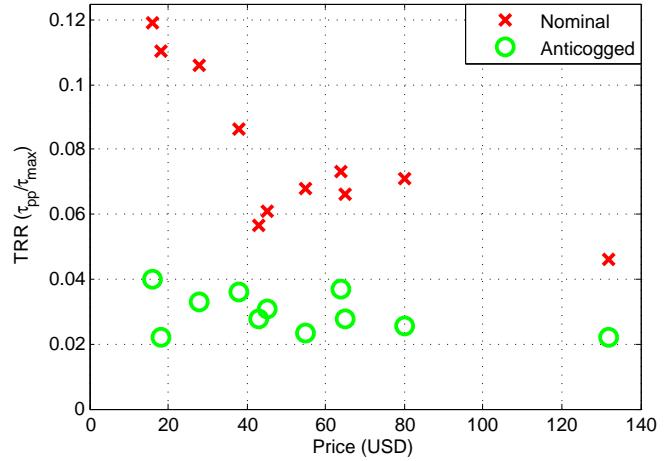


Fig. 1: Nominal and compensated torque ripple ratio vs. price

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 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). Torque ripple is shown in the figure as the torque ripple ratio (TRR , detailed in Equation 9). 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.

A. 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 [5] [6]. In a PMSM, this is the primary source of torque production, having the largest DC component¹. 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.

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 [7]. 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 [8]. 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.

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° 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° 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 [9].

¹When referring to DC components or DC signals, the authors are referring to the non-oscillating offset components in the frequency domain, rather than current.

B. 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 [5] [6] [9] [10] [11]. However, [12] and [13] use current feedback, while Kim and Ha [14] use speed feedback at low speeds for online estimation. In practice, speed control loops and estimation have limited success in minimizing torque ripple at higher speeds due to measurement delays.

Petrovic et al. [7] 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 [5] [10], 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. [8] 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. Torque ripple minimization is either incomplete when using current sensing methods or is prohibitively expensive when using an external torque sensor. However, as we will show, it is possible to measure torque ripple via position or velocity feedback without measuring torque directly. Observing speed changes (i.e. accelerations) and controlling speed to minimize ripple may not work at high operational speeds, but monitoring speed and its ripple at low nominal speeds is comparatively simple. Data gathered at low speeds can be applied at high speeds open loop with notable results. 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 only torque perturbations. Therefore, position error under position control can be used to map cogging torque and friction torque.

This work will use cogging torque waveforms that 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 [5] [10].

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

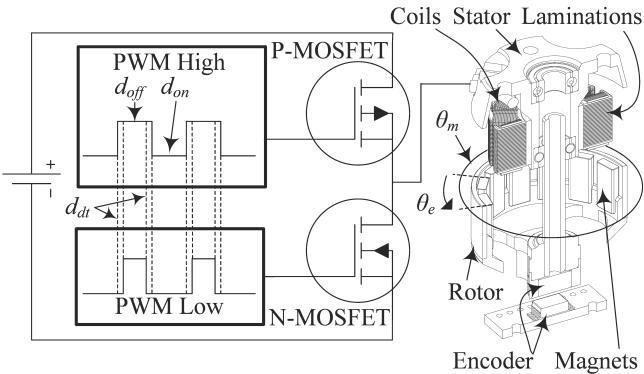


Fig. 2: Diagram of a single half H-bridge inverter connected to one of three phases of a sectioned motor.

II. 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.

A. Assumptions

This paper makes the following assumptions:

- 1) Mutual torque varies linearly with current.
- 2) Each motor winding has equal resistance and inductance.
- 3) A half-H bridge inverter is used to control each phase.
- 4) 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.
- 5) The PWM frequency, motor inductance, and motor inertia are all sufficiently high that the inverter switching does not induce a torque ripple.
- 6) Cogging and friction torque ripples are time-invariant.
- 7) The position sensor has a sufficiently high resolution.

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° added to the electrical position, θ_e . The conversion between electrical position and mechanical position, θ_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. With regards to the controller, the motor model can be represented by:

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

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.

B. Waveform Collection

Two methods of collecting the torque ripple waveform were explored. Both exploit the fact that cogging torque is visible from the mechanical state, i.e. position and speed of the rotor. Using standard position encoders to collect data maintains a low cost to the anticogged system.

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

```

1) Position Based: The position based collection of the current or voltage waveform 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 positions:

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

with encoder positions, i , in monotonically increasing order. For a motor with continuous rotation, the minimum encoder position, $\theta_{m,min}$, equals $\theta_{m,max}$, the maximum encoder position, after a full rotation and i spans the full encoder count range. At each commanded position i , measurements are recorded including:

$$\theta_{m,act,i}, \quad d_i, \quad V_{sup,i}, \quad I_i$$

which are the actual position, applied PWM duty cycle in Per Unit (PU or %/100), supply voltage, and current, respectively. 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 $V_{app} = IR$. When

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

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 III-A. Note the reverse is significantly different.

2) Acceleration Based: Algorithm 2 is used to collect the current or voltage waveform based on motor accelerations. 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

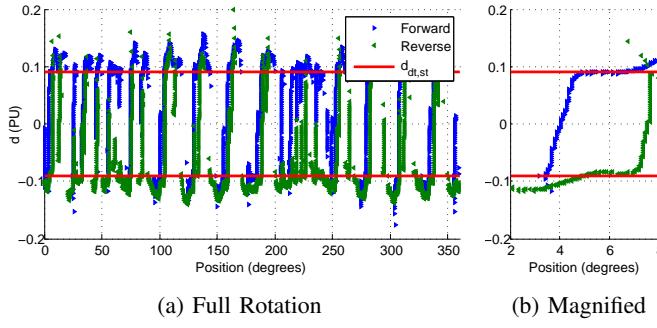


Fig. 3: Position method collected data from motor M4 in Table I described in Section III-A. (a) A full 360° dataset with forward and backward trials and (b) a magnified section.

lowest duty cycle that starts the motor and allows continuous rotation is $d_{dt,st,cg_{max}}$, 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 II-C). 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 $d_{dt,st,cg_{max}}$ for some amount of time or number of revolutions, then d_{min} is commanded. The position sensor measures position, θ_m , and its derivative gives speed, $\dot{\theta}_m$, for each encoder location, m . Repeating this process in the opposite direction yields similar results.

Algorithm 2 Acceleration Based Waveform Collection

```

 $d_{dt,st,cg_{max}} \leftarrow 0$ 
while  $\theta_m \neq 0 \quad \forall \theta_{m,i}$  do
    if  $\dot{\theta}_m = 0$  then
         $d_{dt,st,cg_{max}} \leftarrow d_{dt,st,cg_{max}} + \min \Delta d$ 
    end if
    Command  $d_{dt,st,cg_{max}}$ 
end while
 $d_{min} \leftarrow d_{dt,st,cg_{max}} - \min \Delta d$ 
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  $d_{dt,st,cg_{max}}$ 
Wait
Command  $d_{min}$ 
Wait
 $j \leftarrow 0$ 
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

```

C. Waveform Analysis

For cogging compensation, the data collected in Algorithms 1 and 2 must be converted to voltage or current waveforms, $I_{cg,i}$ or $V_{cg,i}$. Fast Fourier Transforms (FFTs) and bi-cubic splines have been used for fitting similar waveforms in order to fill gaps in collected data since collection at every encoder position is not guaranteed [15]. 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. Supply level voltages are produced when only high or 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 time². 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, $d_{cg,i}$ and $I_{cg,i}$. 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 currents, voltage is linear with current, I_{st} , and thus is linear with torque. However, because deadtime is a time, it is compensated

²Deadtime refers to only the time that neither transistor is conducting, and not deadzone, the range of mechanical position slop.

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 compensated, the voltage or current waveforms can be extracted. When using the position method, $I_{cg,i}$ falls out from Algorithm 3 and $V_{cg,i}$ can be found using $d_{cg,i}$ as d in Equation 3. When using the acceleration method, the accelerations are found by taking the time derivative of the FFT fitted speeds, $\ddot{\theta}_{m,i} = \frac{d\dot{\theta}_{m,i}}{dt_j}$. Noting that the rotor inertia, J , is constant, the cogging torque is then:

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

The motor parameters can be used to find the mapping between $\tau_{cg,i}$, $I_{cg,i}$, and $V_{cg,i}$. Alternatively, $d_{dt,st,cg_{max}}$ with Equation 3 can be used to scale the acceleration waveform to find $V_{cg,i}$.

D. Waveform Suppression

For either current or voltage control, FFTs are fitted to the data with respect to mechanical position as mentioned in Section II-C. 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 $V_{cg,i}$ or $I_{cg,i}$. 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:

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

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

or

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

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, θ_m , is a mechanical value and thus captures all torque ripple sources, these additional torques are already compensated for within the algorithm.

III. 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 [16]. The arm specifications includes a planar 2 DOF subset of the Geometric Touch workspace. This workspace is advertised as rectangular area (160×120) mm. However, it is a polar device with workspace measurements between $100 \text{ mm} < \text{radius} < 270 \text{ mm}$ and 90° in angler range. The maximum continuous force output is 880 N mm.

Algorithm 3 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
  end for
   $d_{dt,st,i} \leftarrow \frac{d_{max,i} - d_{min,i}}{2}$ 
   $d_{cg,i} \leftarrow \frac{d_{max,i} + d_{min,i}}{2}$ 
   $I_{st,i} \leftarrow \frac{I_{max,i} - I_{min,i}}{2}$ 
   $I_{cg,i} \leftarrow \frac{I_{max,i} + I_{min,i}}{2}$ 
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
   $d_{dt,st,k_{temp}} \leftarrow d_{dt,st,i}$ 
   $k \leftarrow k + 1$ 
end for
 $d_{dt,st} = \bar{d}_{dt,st,k_{temp}}$ 
 $k \leftarrow 0$ 
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
 $V_{st} \leftarrow \bar{d}_{st,i}\bar{V}_{sup,i}$ 
 $I_{st} \leftarrow \bar{I}_{st,i}$ 
 $d_{dt} \leftarrow d_{dt,st} - \bar{d}_{st,i}$ 

```

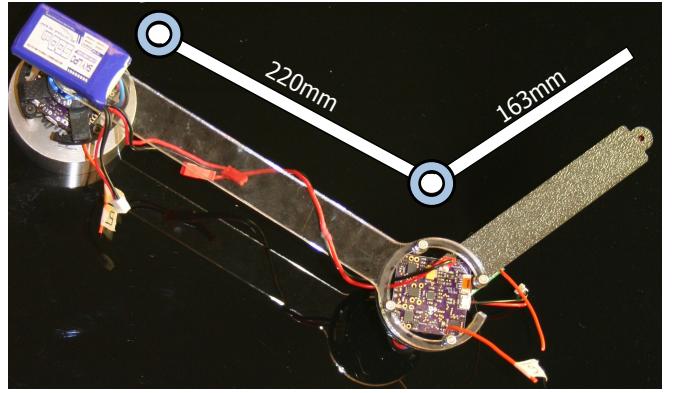


Fig. 4: Top view of the robotic arm.

Figure 4 shows the two link serial chain used for final validation. 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 with an effective envelope of $57 \text{ mm} < \text{radius} < 383 \text{ mm}$, and 360° angular range which encompasses the required $(160 \times 120) \text{ mm}$ 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 with parallel links and 0.59 mm with perpendicular links, which 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° . The maximum applicable force occurs with the shorter lever arm creating the largest static force. This gives a target max motor torque of $0.88 \text{ N} \cdot 163 \text{ mm} = 143 \text{ N mm}$.

A. 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 \$2.91 USD using the AS5115.

Although the PWM frequency of 50 kHz is excessive in position control applications, this frequency is required for high speed rotation. The same setup can be used to spin motors at over 22,000 RPM. At this speed an encoder count greater than 256 may be excessive. The combination high PWM frequency and high encoder resolution allows for smooth commutation from standstill to high speeds with a seamless transition and high fidelity waveform generation. The final version of the arm uses this setup, except an updated motor driver controls phase voltages at 10 kHz with a 1000 count PWM resolution.

For validating the proposed acceleration and position waveform generation methods, a third method is used to provide ground truth, experimentally determining torque ripple. It uses an external torque sensor, sampled while performing the acceleration method. The motor's datasheet values translate torque to voltage and current for this third method.

Value	Unit	M1	M2	M3	M4	M5	M6
R	$m\Omega$	1030	260	29	220	48	400
K_v	rpm/V	285	740	1400	710	1000	750
I_0	mA	183	550	1200	600	1700	600
V_0	V	24	10	10	10	8.4	10
P	Poles	16	14	14	14	14	14
Mass	g	110	56	90	60	142	54
τ_{max}	Nmm	77.7	90.3	239	134	363	127
Cost	USD	132	44.99	18	16	28	55
Dia	mm	42.8	28	37	28	38	28
Length	mm	21.3	29	30	30	44	30
τ_{pp} nom	Nmm	3.6	5.5	26.3	16.0	38.4	8.7
τ_{pp} pos	Nmm	1.7	2.8	21.5	9.2	19.7	4.2
τ_{pp} acc	Nmm	1.7	3.9	5.3	5.4	12.0	3.0
Reduction	%	53	49	80	66	69	65
τ_{res}	Nmm	0.54	0.83	3.9	1.02	3.3	0.53
d_{dt}	PU	.072	.082	.089	.082	.080	.090
V_{st}	mV	0	26.6	12.7	42.1	89.6	117

TABLE I: Motors and Results of Anticogging.

M1 is a Maxon EC 45 251601.

M2 is an E-flite Park 400 EFLM1300.

M3 is an Exceed RC Rocket 86MB83.

M4 is an Exceed RC Rocket 86MA10.

M5 is a Turnigy Sk3542.

M6 is an ElectriFly Rimfire GPMG4555.

The controller and driver are connected to eleven motors, six of which are indicated in Table I. The motors are fixed to an ATI Industrial Automation Nano17 six-axis force and torque transducer with 1/64 N mm resolution. Values of θ , $\dot{\theta}$, V_{sup} , and d are read at 1 kHz by the controller, while torque and current are read at 20 kHz in MATLAB.

B. Results

A common metric of torque ripple is the torque ripple factor (TRF) [9] [8]. The equation for TRF is:

$$TRF = \frac{\tau_{pp}}{\tau_0} \quad (8)$$

where τ_{pp} is the peak to peak torque variation and τ_0 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 τ_0 . Since cogging torque is independent of current and thus τ_0 , TRF is not constant and is less useful. TRF is infinite for all motors at zero applied torque, but there is still torque ripple from cogging. In place of τ_0 , a divisor that remains constant for each motor is proposed as *Torque Ripple Ratio* or *TRR*, defined as follows:

$$TRR = \frac{\tau_{pp}}{\tau_{max}} \quad (9)$$

where τ_{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 5 shows a plot of the before and after results of applying anticogging to the 11 tested motors. A line fit shows

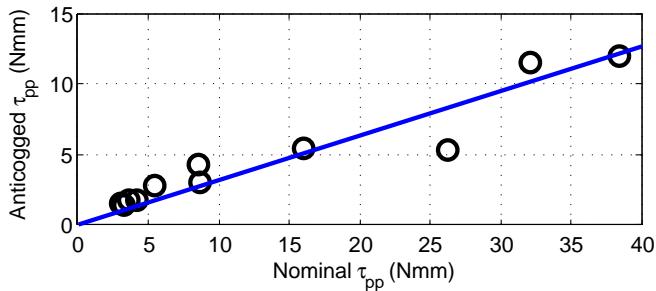


Fig. 5: Torque ripple after anticogging versus torque ripple before anticogging for eleven tested motors. Fit line is $y = 0.3139x$ with an $R^2 = 0.8922$.

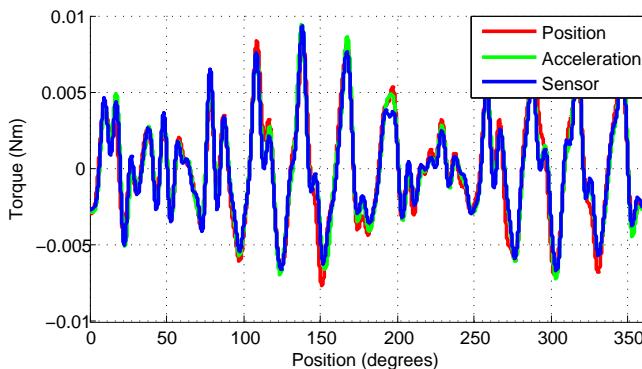


Fig. 6: Fitted position, acceleration, and torque data versus position on an Exceed RC 86MA10 motor. Voltages are converted to torques using motor datasheet parameters.

a 69 % average reduction in torque ripple. Table I shows the details for a subset of the motors from Figure 5. $\tau_{pp\ nom}$ is the nominal peak to peak cogging torque of the motor. $\tau_{pp\ pos}$ and $\tau_{pp\ acc}$ are the peak to peak cogging torques after applying the position method and the acceleration method, respectively.

A metric that describes the value of a motor is $\tau_{pp} \times \text{cost}$. From the results in Table I, 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.

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 7 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. The effect of cogging on the end point position is clearly evident in the figure. The 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.

IV. DISCUSSION

A. Comparison of Methods

Verifying that both methods of cogging characterization map the torque ripple accurately is crucial. Figure 6 displays both methods as well as the ground truth from the external torque sensor. From the plot, the reader can see that all three methods are in agreement in shape, while the position method was slightly less accurate, as indicated in Table I. All successful characterizations have a RMS error of $< 1 \text{ N mm}$.

It is mentioned in Section I-B 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 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 signal³ 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 I. 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 $V_{st} = 60 \text{ mV}$ at the tested location. The discrepancy between these values and those in Table I 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 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. Also, the position method tends to predict higher torques, resulting in less iterations, but increased overcompensation when iterating.

With a sufficient quality waveform map loaded into the driver's onboard memory, the fidelity of the output wave-

³When referring to DC components or DC signals, the authors are referring to the non-oscillating offset components in the frequency domain, rather than current.

form 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 0.01667 V increments. Converting this voltage increment into torque increments for each motor using datasheet parameters, $\tau = K_T I$, and $V = IR$, gives the values indicated by τ_{res} in Table I. It can be seen that the resolutions are on the same order of magnitude as the anticogged τ_{pp} , between 1 and 5 counts across the full range. This indicates that PWM resolution is the limiting factor of torque ripple reduction in this dataset. A supplementary test with a 600 count PWM, which is double previous tests, was performed on motor M5 from Table I. The τ_{pp} is further reduced to 7.0 N mm, which is a 42% reduction from the lower resolution anticogging trial and an 82% reduction from nominal, consistent with PWM as a leading factor in determining the amount of possible reduction.

B. Arm Test Results

The results in the previous paragraph guided the design of the updated motor driver used in the robotic arm and is described at the end of Section III. 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 7.

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 when frequencies >5 Hz [17]. Nominal human motions move the arm at 120° in 1 second, that would correspond to approximately 5 Hz as the dominant frequency in Figure 6 over 120°. Faster motions would result in higher frequencies to which humans are much more sensitive.

V. 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.

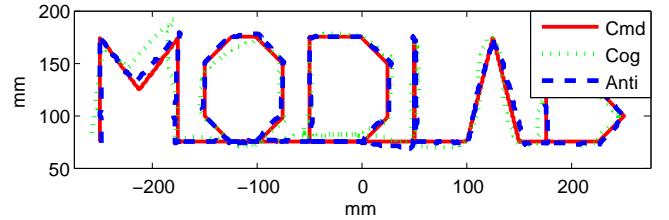


Fig. 7: Trajectory of robotic arm with and without anticogging.

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 80% have been seen across the 11 tested motors.

In many cases, cheap 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 I has slightly better mass and max torque, but has double the uncompensated torque ripple and is 30% of the cost of the M6 motor. Using the acceleration compensation method, the M4 τ_{pp} is reduced to 5.4 N mm which is significantly better than the M6 uncompensated τ_{pp} of 8.7 N mm. Similar comparisons can be made with most of the other tested motors.

There are several ways to improve the performance of the proposed anticogging. Increasing PWM resolution is one of the most straightforward mechanisms. Decreasing the PWM frequency will allow higher resolution at the cost of a loss of resolution at high speeds. Online waveform mapping and adjustment while performing arbitrary tasks might be useful for motor properties that vary over time, though it's likely these characteristics will change very slowly. Upon successful torque ripple minimization, Hanselman [11] 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

The authors would like to acknowledge the support of NSF contract number 1138847 and Willow Garage. We would also 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.

REFERENCES

- [1] G.J. Liu and A.A. Goldenberg. Robust control of robot manipulators incorporating motor dynamics. In *Intelligent Robots and Systems' 93, IROS'93. Proceedings of the 1993 IEEE/RSJ International Conference on*, volume 1, pages 68–75. IEEE, 1993.
- [2] W.S. Newman and J.J. Patel. Experiments in torque control of the AdeptOne robot. In *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, pages 1867–1872. IEEE, 1991.
- [3] 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.
- [4] Vincent Hayward and Karon E. MacLean. Do It Yourself Haptics: Part I. *IEEE Robotics and Automation Magazine*, pages 88–104, December 2007.
- [5] 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.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] 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.
- [10] 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.
- [11] 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.
- [12] 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.
- [13] 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.
- [14] 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, Jul 1999. ISSN 1063-6536. doi: 10.1109/87.772162.
- [15] 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, jun-3 jul 1992. doi: 10.1109/PESC.1992.254794.
- [16] Geomagic. Geomagic Touch Specifications, January 2014.
- [17] 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.