Optimal Current Waveforms for Brushless Permanent Magnet Motors

Nicholas Moehle* Stephen Boyd†

April 14, 2014

Abstract

In this paper we give energy-optimal excitation current waveforms for a permanent magnet synchronous motor that result in a desired average torque. Our formulation generalizes previous work by including a general back-EMF waveform, voltage and current limits, an arbitrary phase winding connection, a simple eddy current loss model, and a trade-off between power loss and torque ripple. Determining the optimal current waveforms requires solving a small convex optimization problem. We give a fast algorithm to find the optimal current waveforms in around 400μs; changes in required torque can be handled in around 200μs even on low-cost processors. We show that for an ideal motor with sinusoidal back-EMF, the optimal current waveforms increase efficiency by several percent in the constant power region. We also show that, for our model parameters, a sinusoidal back-EMF gives no performance benefit over a trapezoidal back-EMF, even in the constant power region, provided the optimal current waveforms are used. Another advantage of on-line optimization is the ability to adapt in real time to changes in the model or requirements, such as changes in resistance as winding temperature varies, or even gross changes like the failure of one winding.

*Mechanical Engineering Department, Stanford University. moehle@stanford.edu

†Electrical Engineering Department, Stanford University. boyd@stanford.edu

Contents

1 Introduction **3**
- 1.1 Related work . . . 4
- 1.2 Outline . . . 5

2 Model **5**
- 2.1 Dynamics . . . 6
- 2.2 Inverter . . . 7
- 2.3 Magnetic saturation . . . 7
- 2.4 Torque . . . 8
- 2.5 Power loss . . . 8

3 Torque control problem **8**
- 3.1 Optimal torque control . . . 8
- 3.2 Variations . . . 9
- 3.3 Symmetry . . . 10

4 Solution **11**
- 4.1 Discretization . . . 11
- 4.2 Interior-point methods . . . 12
- 4.3 Operator splitting . . . 12
- 4.4 Solve times . . . 13

5 Implementation **14**

6 Example **15**
- 6.1 Sinusoidal vs. rectangular back-EMF . . . 16
- 6.2 Optimal vs. sinusoidal current waveforms . . . 16
- 6.3 Open-phase fault . . . 18

7 Conclusion **18**

1 Introduction

We consider the problem of controlling an AC permanent magnet synchronous motor (PMSM) by choosing phase winding current waveforms that produce smooth output torque. Traditionally, the problem is solved differently depending on the type of motor: if the rotor magnets induce an counter-electromotive force (back-EMF) in the phase windings that is a sinusoidal function of rotor position (*i.e.*, the motor has a *sinusoidal back-EMF waveform*), then sinusoidal current waveforms are used; if the induced back-EMF is instead a trapezoidal function (*i.e.*, the motor has a *trapezoidal back-EMF waveform*), then rectangular current waveforms are used. Ideally, both of these schemes produce smooth output torque. In reality, however, motors with perfectly shaped back-EMF waveforms are difficult to produce, and rectangular phase current waveforms are impossible to realize due to phase winding self-inductance.

We therefore address the problem of choosing drive current waveforms that achieve a desired average torque while minimizing a combination of resistive power loss and RMS torque ripple. We consider motors with general back-EMF waveforms. We assume supply voltage limits and phase current limits due to magnetic saturation. We also include a simple eddy current loss model, which has the effect of penalizing high-frequency harmonics of the current waveforms. Because our formulation can be applied to motors with arbitrary phase connections (including delta, wye, and independently connected phases), we can handle several fault conditions, and we demonstrate operation of a delta-wound, three-phase motor with a single open-phase fault. We also discuss simple variations of our formulation, including alternative definitions of torque ripple (*e.g.*, the range of the torque values, or the mean absolute deviation), or maximum torque problems. We show that the proposed torque control problem, and all proposed variations, are convex optimal control problems, and can be therefore be quickly and reliably solved using convex optimization. We provide a fast iterative algorithm for solving the resulting optimization problem, and we demonstrate that the algorithm can be executed quickly, in well under one millisecond. Furthermore, the iterates of the algorithm provide a very good approximation of the optimal waveforms, even before convergence. Indeed, the first iteration of the algorithm (available in tens of microseconds) produces the optimal current waveforms when voltage and current limits are ignored, which is competitive with evaluating the analytical expressions given in the literature (see below). Within a few tens of iterations, the iterates are typically within a few tenths of a percent of the optimal values, and can be considered converged for all practical purposes. The algorithm can also incorporate new desired torque signals after each iteration, and new iterates can be implemented immediately, which reduces the torque response time to tens of microseconds on a standard processor, or hundreds of microseconds on a low-cost ARM processor, which is around the switching period of common power electronic devices. The inverter bridge voltage waveforms that generate the optimal current waveforms are also computed as a by-product of computing the optimal current waveforms; these open-loop optimal bridge voltages and current waveforms can be used in a closed-loop current control scheme.

The ability to find the optimal current waveforms in real time allows us to change the model parameters on the fly (*e.g.*, changing phase resistance with temperature, or updating

the inverter bus voltage), or change the problem based on operating requirements. For example, an electric vehicle application may require maximizing output torque at some times, thus increasing the performance of the vehicle, and high efficiency and low torque ripple at other times, thus increasing the efficiency (and range) of the vehicle.

We demonstrate our formulation on an example motor. Our method gives the optimal current waveforms in the constant power (*i.e.*, flux weakening) operating region, and we demonstrate that the optimal drive currents are nonsinusoidal in this region, even for motors with sinusoidal back-EMF. We additionally compare the performance of sinusoidal and trapezoidal back-EMF waveforms in this region, and we find that, when driven with the optimal current waveforms, the two types of motors perform quite similarly.

1.1 Related work

Convex optimization. Convex optimization problems can be solved efficiently and reliably using standard techniques [BV04]. Recently, much work has been devoted to solving moderately-sized convex optimization problems quickly (*i.e.*, in milliseconds or microseconds), possibly on embedded platforms, which enables convex-optimization-based control policies to be implemented at kilohertz rates [WB10, OSB13]. In addition, recent advances in automatic code generation for convex optimization [MB10, CPDB13] can significantly reduce the cost and complexity of developing and verifying an embedded solver.

We provide an algorithm based on the alternating direction method of multipliers (ADMM). Details about ADMM can be found in [BPC+11, PB14]. ADMM is is particularly well-suited for real-time optimal control because it typically converges to acceptable accuracy very quickly, and because its simplicity allows for easily verifiable source code. Details about using ADMM for optimal control can be found in [OSB13].

PMSM current waveform optimization. Significant work has been devoted to finding optimal current waveforms for motors with nonsinusoidal back-EMF waveform, and several special cases have been solved, some analytically. Simple characterizations of current waveforms that minimize power loss and produce smooth torque in the absence of other constraints are given in [LHPF86, HD92, WC05]. The authors of [CSW99] extend these results to include a trade-off between RMS torque ripple and power loss by solving a linear least squares problem. An RMS voltage limit is considered by [Han94], who proposes offline solution of the resulting quadratic program and implementation as a lookup table. The authors of [ABH01, ABH03] instead introduce a current saturation limit, and analytically solve the Karush-Kuhn-Tucker conditions to find the optimal current waveforms that produce smooth torque. Several of the aforementioned results can be extended to apply if one or more phases are in an open fault condition; explicit derivation of the optimal waveforms in this case can be found in [BMDL13] and references therein (which neglect voltage and current limits). The authors of [YWWL04] take a different approach, showing that minimum power loss required to achieve a desired torque, without regard for torque ripple, is attained when the current waveform is proportional to the back-EMF waveform.

Field oriented control. For a motor with sinusoidal back-EMF waveforms and no voltage limits, sinusoidal current waveforms minimize power loss while also producing no torque ripple. In this case, it is convenient to formulate the control problem using Park's transformation into the d–q reference frame, which separates the current producing component of the current waveform from the field-weakening component, resulting in a *field-oriented control* framework [GLN80, HM94],

Within this framework, two general techniques can be used to minimize power loss in steady-state, including RMS voltage constraints In loss-model control, a model of the power loss is used to determine the sinusoidal currents that minimize power loss in steady-state at a given operating point. Early work in this area focused on finding analytical expressions for the optimal current for a simple loss models [CCL08, MTTH94]. More recent approaches have addressed nonlinearities due to magnetic saturation, by storing a lookup table of optimal d- and q-axis current components [LNCK07], or computing them in real time using Newton's method. [JSHR06]. *Search control* techniques instead involve actively searching for the optimal current d- and q-axis current components by directly iterating on the motor itself [VJR99, CN88].

Model predictive control. As an alternative to seeking the optimal steady-state currents, model predictive control has recently been proposed to improve dynamic response. Most formulations involve solving a convex optimization problem to determine the inverter voltage signal, which can either be done offline and implemented as a lookup table, as in [WsCY$^+$14, MDM09, BKKP11, BBPZ09] or online in real time, as in [SDK12]. A different model predictive control scheme is proposed in [Gey11], who considers the discrete switching states of the inverter. Using various heuristic strategies, they are able to compute a reasonable sequence of inverter switching states.

1.2 Outline

In §2, we introduce our model of the PMSM. In §3, we formally introduce the torque control problem, and we list several variations of the base problem. We then explore the types of symmetry that a PMSM typically exhibits, and we show how to use symmetry to reduce the complexity of the torque control problem. In §4, we demonstrate the solution of the torque control problem, and we give a fast operator splitting algorithm. In §5, we discuss how to implement the solution in real time, possibly on an embedded processor. In §6, we show the optimal waveforms for an example motor, which we use to compare rectangular and sinusoidal back-EMF waveforms, and to compare the optimal waveforms with sinusoidal current waveforms.

2 Model

The model describes a three-phase, permanent magnet synchronous motor, shown in figure 1. The rotor, which is nonconductive and contains permanent magnets, has angular position θ

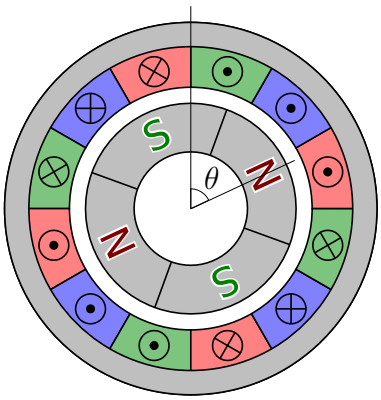

Figure 1: Schematic of permanent magnet synchronous motor. The rotor has angular position θ. Symbols $\otimes$ and $\odot$ represent the direction of axial windings. Different phase windings are different colors.

and angular velocity ω; we assume ω is constant. The stator contains three circuits, called *phase windings*, with currents i_a, i_b, and i_c, and voltages v_a, v_b, and v_c. The phase currents induce eddy currents in the stator iron; we consider induced eddy currents separately for each phase, which we call j_a, j_b, and j_c. The motor is driven by a voltage-source, triple-half-bridge inverter with bridge voltages v_U, v_V, and v_W. The output torque is τ.

We will assume that i_a, i_b, i_c, v_a, v_b, v_c, j_a, j_b, j_c, v_U, v_V, v_W, and τ are 2π-periodic functions of θ. We use a prime ($'$) to denote differentiation of these functions with respect to θ. To lighten notation, we often drop explicit dependence on θ.

2.1 Dynamics

The circuit dynamics of the phase windings are

$$\begin{aligned} v_\mathrm{a} &= Ri_\mathrm{a} + \omega(Li_\mathrm{a}' + Mi_\mathrm{b}' + Mi_\mathrm{c}' + \tilde{M}j_\mathrm{a}' + k_\mathrm{a}), \\ v_\mathrm{b} &= Ri_\mathrm{b} + \omega(Mi_\mathrm{a}' + Li_\mathrm{b}' + Mi_\mathrm{c}' + \tilde{M}j_\mathrm{b}' + k_\mathrm{b}), \\ v_\mathrm{c} &= Ri_\mathrm{c} + \omega(Mi_\mathrm{a}' + Mi_\mathrm{b}' + Li_\mathrm{c}' + \tilde{M}j_\mathrm{c}' + k_\mathrm{c}), \end{aligned} \tag{1}$$

where R is the phase resistance, L is the phase self inductance, M is the mutual inductance between phases, and $\tilde{M}$ is the mutual inductance between a phase and the corresponding eddy current. The back-EMF waveforms k_a, k_b, and k_c are 2π-periodic functions of θ. The eddy current circuits are modeled as RL (resistance-inductance) circuits, with dynamics

$$\begin{aligned} 0 &= \tilde{R}j_\mathrm{a} + \omega(\tilde{L}j_\mathrm{a}' + \tilde{M}i_\mathrm{a}'), \\ 0 &= \tilde{R}j_\mathrm{b} + \omega(\tilde{L}j_\mathrm{b}' + \tilde{M}i_\mathrm{b}'), \\ 0 &= \tilde{R}j_\mathrm{c} + \omega(\tilde{L}j_\mathrm{c}' + \tilde{M}i_\mathrm{c}'), \end{aligned} \tag{2}$$

where $\tilde{R}$ and $\tilde{L}$ are respectively the eddy circuit resistance and self inductance.

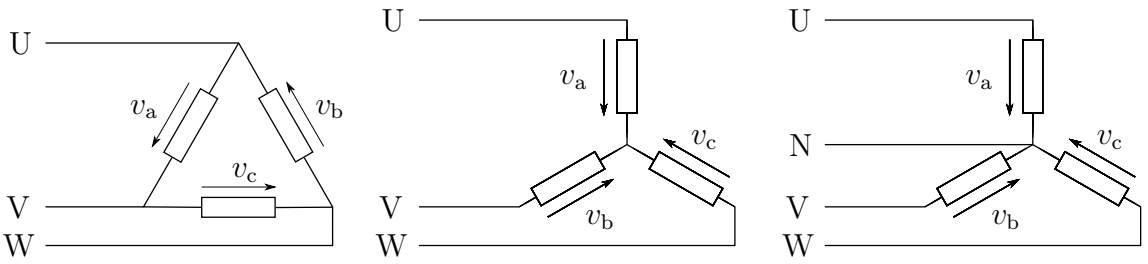

Figure 2: Delta, wye, and independently driven phase connections.

2.2 Inverter

The inverter bridge voltages satisfy

$$|v_{\mathrm{U}}| \leq (1/2)V_{\mathrm{dc}}, \qquad |v_{\mathrm{V}}| \leq (1/2)V_{\mathrm{dc}}, \qquad |v_{\mathrm{W}}| \leq (1/2)V_{\mathrm{dc}}, \tag{3}$$

where V_{dc} is the constant DC bus voltage (figure 2). We consider three winding configurations, The relation between the bridge voltages and the phase voltages depends on the winding connection, and are given below. depends on the winding connection, and are given below.

- Independent phases:

$$v_{\mathrm{a}} = v_{\mathrm{U}}, \qquad v_{\mathrm{b}} = v_{\mathrm{V}}, \qquad v_{\mathrm{c}} = v_{\mathrm{W}}. \tag{4}$$

- Delta connection:

$$v_{\mathrm{a}} = v_{\mathrm{U}} - v_{\mathrm{V}}, \qquad v_{\mathrm{b}} = v_{\mathrm{V}} - v_{\mathrm{W}}, \qquad v_{\mathrm{c}} = v_{\mathrm{W}} - v_{\mathrm{U}}. \tag{5}$$

- Wye connection:

$$v_{\mathrm{a}} - v_{\mathrm{b}} = v_{\mathrm{U}} - v_{\mathrm{V}}, \qquad v_{\mathrm{b}} - v_{\mathrm{c}} = v_{\mathrm{V}} - v_{\mathrm{W}}, \qquad v_{\mathrm{c}} - v_{\mathrm{a}} = v_{\mathrm{W}} - v_{\mathrm{U}}, \tag{6}$$

  as well as Kirchoff's current law for the center node:

$$i_{\mathrm{a}} + i_{\mathrm{b}} + i_{\mathrm{c}} = 0. \tag{7}$$

Other configurations are possible. For example, a six-half-bridge inverter can arbitrarily assign voltages to each side of the three phases (with each bridge voltage between $-(1/2)V_{\mathrm{dc}}$ and $(1/2)V_{\mathrm{dc}}$).

2.3 Magnetic saturation

We assume that the phase currents are maintained within the following magnetic saturation limits:

$$|i_{\mathrm{a}}| \leq i^{\max}, \qquad |i_{\mathrm{b}}| \leq i^{\max}, \qquad |i_{\mathrm{c}}| \leq i^{\max}. \tag{8}$$

2.4 Torque

The total output torque is

$$\tau = k_{\mathrm{a}} i_{\mathrm{a}} + k_{\mathrm{b}} i_{\mathrm{b}} + k_{\mathrm{c}} i_{\mathrm{c}} + \tau_{\mathrm{cog}}, \tag{9}$$

where the cogging torque τ_{cog} is a 2π-periodic function of θ. The average torque over one cycle is

$$\overline{\tau} = \frac{1}{2\pi} \int_0^{2\pi} \tau \; d\theta.$$

The RMS torque ripple is

$$r_{\mathrm{rms}} = \sqrt{\frac{1}{2\pi} \int_0^{2\pi} (\tau - \overline{\tau})^2 \; d\theta},$$

and the relative torque ripple is

$$r_{\mathrm{rms,rel}} = \frac{r_{\mathrm{rms}}}{\overline{\tau}}.$$

2.5 Power loss

The power loss is the average resistive loss from all phase currents and eddy currents over one cycle:

$$P_{\mathrm{loss}} = \frac{1}{2\pi} \int_0^{2\pi} \left(R(i_{\mathrm{a}}^2 + i_{\mathrm{b}}^2 + i_{\mathrm{c}}^2) + \tilde{R}(j_{\mathrm{a}}^2 + j_{\mathrm{b}}^2 + j_{\mathrm{c}}^2) \right) d\theta.$$

The relative power loss is

$$P_{\mathrm{loss,rel}} = \frac{P_{\mathrm{loss}}}{\overline{\tau}\omega}.$$

and the efficiency is $\eta = 1 - P_{\mathrm{loss}}$. This assumes that $\overline{\tau}\omega > 0$, *i.e.*, the mechanical output power is positive. When the motor is used as a generator, or for regenerative braking, the denominator can be replaced by its absolute value.

3 Torque control problem

3.1 Optimal torque control

We choose the phase voltages, phase currents, and eddy currents that achieve a desired average torque while minimizing the average power loss and torque ripple. The optimal torque control problem is

$$\begin{array}{ll} \text{minimize} & P_{\mathrm{loss}} + \lambda r_{\mathrm{rms}}^2 \\ \text{subject to} & \overline{\tau} = \tau^{\mathrm{des}}, \\ & \text{equations (1), (2), (3), (8), (9),} \\ & \text{and one of (4), (5), or (6)–(7).} \end{array} \tag{10}$$

The parameters are the trade-off parameter $\lambda \geq 0$, The circuit parameters R, L, and M, the eddy circuit parameters $\tilde{R}$, $\tilde{L}$, and $\tilde{M}$, the current and voltage limits $i^{\max}$ and V_{dc}, and

ω and τ^{des}. The variables are the 2π-periodic functions i_{a}, i_{b}, i_{c}, j_{a}, j_{b}, j_{c}, v_{a}, v_{b}, v_{c}, v_{U}, v_{V}, v_{W}, and τ. The constraints include the dynamics, current and voltage limits, torque-current relation, and one set of winding constraints.

Problem (10) is an infinite-dimensional (convex) quadratic program, since the variables to be determined are periodic functions. Once we discretize the variables, it can be (approximately) solved quickly and reliably using standard methods [BV04].

3.2 Variations

We list some variations of (10) that also result in convex optimization problems.

Power loss and torque ripple constraints. We can limit the acceptable power loss and torque ripple by adding the constraints

$$P_{\mathrm{loss}} \leq P_{\mathrm{loss}}^{\mathrm{max}}, \qquad r_{\mathrm{rms}} \leq r_{\mathrm{rms}}^{\mathrm{max}}. \tag{11}$$

Alternatively, we can use the relative values of power loss or relative torque ripple in the objective or in (11).

Maximum torque. We can set up a maximum torque problem. To do this, we remove the $\overline{\tau} = \tau^{\mathrm{des}}$ constraint, and use $\overline{\tau}$ as a variable. Instead of minimizing the power loss and torque ripple, we instead maximize $\overline{\tau}$. Other constraints may be added; for example, a power loss constraint can be used to obtain the maximum sustainable torque subject to heat dissipation limits.

Alternative ripple definitions. We can use other definitions for the torque ripple, such as

$$r_{\mathrm{range}} = \sup_{\theta} \tau(\theta) - \inf_{\theta} \tau(\theta)$$

or

$$r_{\mathrm{abs}} = \frac{1}{2\pi} \int_0^{2\pi} |\tau - \overline{\tau}| \, d\theta.$$

These are both convex, nonquadratic functionals of τ.

Mitigation of harmonics. We can establish magnitude constraints or penalties on specified harmonics of the currents or torque (*e.g.*, to avoid a known mechanical resonance).

Open-phase fault. We can continue operation with one inoperable phase winding by eliminating the relevant equations and variables from the model.

3.3 Symmetry

Although the variables and parameters of (10) are fully defined by their values over the interval $[0, 2\pi]$, we can use the following three assumptions to make this interval shorter, thus reducing the complexity of a discretized version of (10). This exploitation of symmetry does not change the solution; it merely results in a smaller problem that has to be solved. We also note that in some cases the symmetry assumptions do not hold, in which case, we obviously cannot exploit symmetry. Examples of this include motors intentionally designed without symmetry, or when a winding in an otherwise symmetric motor has failed.

Pole symmetry. We assume the rotor has N_p pole pairs (figure 3), *i.e.*, k_a, k_b, k_c, and τ_cog are $2\pi/N_\mathrm{p}$-periodic. Consequently, there is a solution of (10) in which the optimal variables are also $2\pi/N_\mathrm{p}$-periodic.

Half-wave symmetry. We further assume that k_a, k_b, k_c and τ_cog are half-wave symmetric with period $2\pi/N_\mathrm{p}$ (*e.g.*, $k_\mathrm{a}(\theta) = -k_\mathrm{a}(\theta + \pi/N_\mathrm{p})$). This implies there exists a solution of (10) in which the optimal variables are also half-wave symmetric with period $2\pi/N_\mathrm{p}$.

Phase symmetry. We assume the stator windings are displaced by $2\pi/(3N_\mathrm{p})$ radians from each other, so the back-EMF waveform in the second and third phases are shifted versions of the first:

$$k_\mathrm{b}(\theta) = k_\mathrm{a}\left(\theta + \frac{2\pi}{3N_\mathrm{p}}\right), \qquad k_\mathrm{c}(\theta) = k_\mathrm{a}\left(\theta - \frac{2\pi}{3N_\mathrm{p}}\right). \tag{12}$$

and τ_cog is $2\pi/(3N_\mathrm{p})$-periodic. This implies there exists a solution of the original problem in which the optimal variables are shifted versions of each other:

$$i_\mathrm{b}(\theta) = i_\mathrm{a}\left(\theta + \frac{2\pi}{3N_\mathrm{p}}\right), \qquad i_\mathrm{c}(\theta) = i_\mathrm{a}\left(\theta - \frac{2\pi}{3N_\mathrm{p}}\right), \tag{13}$$

with similar shift relations for the voltage and eddy current. These assumptions, combined with (9) and (13), imply τ is $2\pi/(3N_\mathrm{p})$-periodic.

Equivalent problem. The symmetry assumptions allow us to form an equivalent problem with the same constraints and objective as (10) in which the variables have domain $[0, \pi/(3N_\mathrm{p})]$. We also add periodicity constraints of the form

$$i_\mathrm{a}(0) = -i_\mathrm{c}\left(\frac{\pi}{3N_\mathrm{p}}\right), \qquad i_\mathrm{b}(0) = -i_\mathrm{a}\left(\frac{\pi}{3N_\mathrm{p}}\right), \qquad i_\mathrm{c}(0) = -i_\mathrm{b}\left(\frac{\pi}{3N_\mathrm{p}}\right),$$

with similar constraints for the voltage and eddy currents, and $\tau(0) = \tau(\pi/(3N_\mathrm{p}))$. The integrands in the definitions of average torque, torque ripple, and power loss are each $\pi/(3N_\mathrm{p})$-periodic; to get equivalent definitions of these values over the appropriate domain, we can integrate over $[0, \pi/(3N_\mathrm{p})]$ instead of $[0, 2\pi]$, and scale the result by 6.

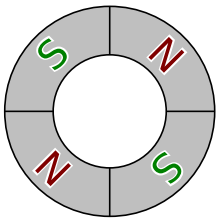

Figure 3: A rotor with pole symmetry ($N_\mathrm{p} = 2$). The magnetic field generated by the permanent magnets is identical at θ and $\theta + \pi$.

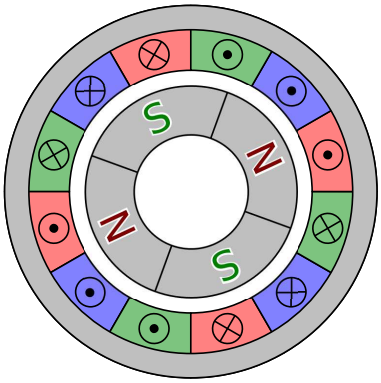

Figure 4: A motor with pole and phase symmetry ($N_\mathrm{p} = 2$).

Any set of optimal variables for the reduced problem is the restriction to $[0, \pi/(3N_\mathrm{p})]$ of some set of optimal variables to the original problem. To reconstruct the values of these variables over the rest of the interval $[0, 2\pi]$, we can use the $2\pi/N_\mathrm{p}$-periodicity and half-wave symmetry of the variables, as well as the phase symmetry shift relations (such as (13)).

4 Solution

4.1 Discretization

After reducing the domain of the variables of (10), we discretize this interval into $N + 1$ grid points, $\theta_0, \ldots, \theta_N$, with $\theta_0 = 0$ and $\theta_N = \pi/(3N_\mathrm{p})$. All pointwise constraints must hold at $\theta_0, \ldots, \theta_{N-1}$, and the periodicity constraints must hold at θ_0 and θ_N. Integration over the interval is replaced by summation from θ_0 to θ_{N-1}, with appropriate scaling. We use a forward-difference approximation to discretize the derivatives in (1) and (2). Implementing this discretization method online is faster than more accurate methods (*e.g.*, first-order hold) and, based on our observations, the inaccuracy does not significantly degrade the quality of the solution.

Finite-dimensional quadratic program. After discretization, problem (10) can be expressed as

$$\begin{array}{ll} \text{minimize} & x^T P x + q^T x + r \\ \text{subject to} & Ax = b, \\ & |x_i| \leq c_i, \quad i = 1 \ldots n. \end{array} \tag{14}$$

The variable $x \in \mathbf{R}^n$ is a vector containing the variables of (10) at $\theta_0 \ldots \theta_N$, and the parameters are the symmetric positive-definite matrix $P \in \mathbf{R}^{n \times n}$, as well as $q \in \mathbf{R}^n$, $A \in \mathbf{R}^{m \times n}$, $b \in \mathbf{R}^m$, $r \in \mathbf{R}$, and $c_i \in \mathbf{R} \cup \{\infty\}$. We note that the parameter matrices P and A are sparse, which can be exploited in a numerical solution.

4.2 Interior-point methods

Problem (14) can be solved using a generic interior-point solver, which involves solving a sequence of linear systems of equations (around 15) of size $(m+n)$. Using a generic quadratic programming solver, the size of these linear systems scales by the cube of the number of grid points. However, a method that exploits the sparse structure of the problem can reduce this to be proportional to the number of grid points. Several software packages are available that can do this, and are capable of running on embedded platforms [MB12, WB10, CPDB13].

4.3 Operator splitting

In addition to interior-point methods, we give an operator splitting algorithm for solving (14). Starting from any $x^{(0)}$, $z^{(0)}$, and $y^{(0)}$, the algorithm generates iterates $x^{(k)}$ according to:

$$x^{(k+1)} = \operatorname*{argmin}_x \left(f(x) + (1/2)\rho \|x + z^{(k)} - y^{(k)}\|^2 \right) \tag{15}$$

$$z^{(k+1)} = \text{sat}(x^{(k+1)} - y^{(k)}, c) \tag{16}$$

$$y^{(k+1)} = y^{(k)} + x^{(k)} - z^{(k)}, \tag{17}$$

where $\rho > 0$ is an algorithm parameter,

$$f(x) = \begin{cases} x^T P x + q^T x & Ax = b \\ \infty & \text{otherwise,} \end{cases}$$

and sat is the vector saturation function, *i.e.*, the ith element of $\text{sat}(x, c)$ is

$$\big(\text{sat}(x, c)\big)_i = \begin{cases} -c_i & x_i < -c_i \\ c_i & x_i > c_i \\ x_i & \text{otherwise.} \end{cases} \tag{18}$$

Note that $x^{(k+1)}$ can be interpreted as a solution of a regularized version of (14) with no inequality constraints (*i.e.*, $c_i = \infty$), and $x^{(k+1)}$ as a projection onto the set of points satisfying the inequality constraints of (14).

Finding the $x^{(k+1)}$ that satisfies (15) can be done by solving

$$\begin{bmatrix} P+\rho I & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x^{(k+1)} \\ \mu \end{bmatrix} = \begin{bmatrix} z^{(k)} - q \\ b \end{bmatrix}. \tag{19}$$

Because each iteration of the splitting algorithm involves solving (19) for a different value of $z^{(k)}$, it is convenient to store the coefficient matrix (the *KKT matrix*) in factored form using a sparse LDL factorization.

Convergence of $x^{(k)}$ to a solution of (14) is guaranteed [BPC+11]. In practice, however, it may be beneficial to iterate over (15)-(17) continuously, updating the (factorized) KKT matrix periodically to reflect the new values of the parameters (in particular, ω). This can be interpreted as a *warm start* for a new instance of (14); because the entries of the KKT matrix change little when updated, we expect the iterates of the previous problem instance to be nearly optimal for the new problem. We therefore can always implement the waveforms contained in the most recent current iterates immediately, because they reflect the current values of all parameters and we expect that they are always very close to optimality. (Indeed, even the first iterate $x^{(0)}$ is the solution of a regularized version of (14) without voltage or current limits.) In addition, because only q and b in (19) depend on τ^{des}, we can update the desired torque after every iteration without refactoring the KKT matrix. Preliminary waveforms that reflect changes in τ^{des} can therefore be implemented after a single linear system solve.

4.4 Solve times

We compare the average solve times of the operator splitting algorithm against two interior-point solvers using the parameter values of the example in §6 (table 1). We used $N = 15$, which corresponds to 270 grid points per cycle (*i.e.*, 90 grid points per pole pair per cycle).

All three algorithms were carried out on a Linux machine with a 3.4 GHz Intel Xeon E31270 processor. In addition, CVXGEN and the operator splitting method were carried out a Raspberry Pi, a $25 computer with a 700 Mhz ARM processor with a floating point unit. Table 1 gives the average solve times for all three algorithms over 1000 uniformly randomly selected torque-speed pairs in the feasible operating region of the motor.

CVX. CVX [GB13, GB08] is a MATLAB-based modeling language for convex optimization. It converts (14) into a second-order cone program, which is solved using the interior-point solver SDPT3 [TTT03, TTT99].

CVXGEN. CVXGEN [MB12] is a code generator for fast convex optimization. It takes a high-level description of a quadratic program and generates a custom interior-point solver, written in C, which is suitable for embedded, real-time optimization. For details on code generation for real-time convex optimal control, see [MWB11, MB10]. CVXGEN was terminated once the duality gap was less than 0.1.

	Intel Xeon	ARM processor
CVX	468.67	—
CVXGEN	3.9	229.8
OS (cold start)	0.12	3.94
OS (warm start)	0.07	2.71

Table 1: Average solve times in milliseconds.

Operator splitting. The operator splitting method was implemented in C. The variables v_{a}, v_{b}, and v_{c} were (analytically) eliminated, resulting in a problem with 150 variables, 107 equality constraints, and 90 inequality constraints. Eq. (19) was solved using the LDL package provided in [Dav05]. The operator splitting method was terminated once $\|x^{(k)} - z^{(k)}\|_\infty$ was less than 0.1 (this represents a 0.1% error for most operating conditions).

If a sequence of similar problem instances is to be solved, we can accelerate convergence of each iteration by initializing the iterates at the solution to the previous problem (called *warm starting*). We provide solve times for the operator splitting method for both cold starting (iterates initialized to zero) and warm starting (iterates initialized to the solution of an identical problem, but with ω uniformly randomly perturbed to a feasible value within 20%).

5 Implementation

Here we collect several ideas for implementing the solution to (10).

Table lookup. Any of the above solvers can be used to to generate a lookup table of optimal waveforms indexed by ω and τ^{des}. This enables the optimal current waveforms to be used on processors with limited computational capability. Some simplifications may be helpful to reduce storage requirements; for example, for motors with wye or independent phase connections, neglecting eddy current, a single current wave shape can be stored for any ω and τ^{des} such that the inequality constraints of (10) are inactive. This wave shape can be scaled to meet the torque requirement.

Real-time optimization. CVXGEN and the operator splitting algorithm are fast enough to compute the optimal waveforms in real time on an embedded system. This allows us to recompute the optimal waveforms after updating the model parameters (*e.g.*, after updating winding resistance with temperature) or after changing the trade-off parameter λ based on performance requirements. We can also change the problem entirely to one of the variations mentioned in §3.2, such as a maximum-torque mode, or continued operation after failure of a phase winding.

Parameter	Value	Unit
R	0.466	Ω
L	3.19	mH
M	-1.31	mH
$\tilde{R}$	3.4	Ω
$\tilde{L}$	2.9	mH
$\tilde{M}$	1.0	mH
V_{dc}	70	V
$i^{\max}$	10	A
N_p	1	—

Table 2: Motor parameters.

Feedback control. If the model is perfectly correct, using the optimal bridge voltage waveforms as PWM signals for the inverter will produce the resulting optimal current waveforms. But the models are of course not perfect, so in practice a closed-loop controller is necessary to ensure accurate tracking of the optimal waveforms For example, a simple state-feedback controller would have the form

$$\begin{bmatrix} v_{\mathrm{U}} \\ v_{\mathrm{V}} \\ v_{\mathrm{W}} \end{bmatrix} = \begin{bmatrix} v_{\mathrm{U}}^{\star} \\ v_{\mathrm{V}}^{\star} \\ v_{\mathrm{W}}^{\star} \end{bmatrix} + K \left(\begin{bmatrix} i_{\mathrm{a}} \\ i_{\mathrm{b}} \\ i_{\mathrm{c}} \end{bmatrix} - \begin{bmatrix} i_{\mathrm{a}}^{\star} \\ i_{\mathrm{b}}^{\star} \\ i_{\mathrm{c}}^{\star} \end{bmatrix} \right) \tag{20}$$

where we use the $\star$ to denote the optimal (reference) values, and $K \in \mathbf{R}^{3\times 3}$ is a controller synthesized based on the circuit dynamics (1). Note that the inclusion of the optimal open-loop bridge voltages will improve dynamic performance compared to a simple tracking controller. Extension to more complex controller architectures (*e.g.*, PI controllers) is beyond the scope of this paper.

6 Example

In this section we demonstrate the optimal waveforms for a numerical example. The values of all scalar model parameters are given in Table 2. The phase resistance, phase inductance, and voltage limits are based on the first example of [LZH05]. The eddy circuit parameters were adjusted to be about one quarter of the total power loss at $\omega = 300$ rad/s, which is the rated speed of the motor. Because the effects of the eddy circuit dynamics are fully described using only two parameters, we arbitrarily set $\tilde{M} = 1$ mH. Cogging torque is neglected. The motor described in [LZH05] has a sinusoidal back-EMF waveform with RMS value of 0.72 V/s. For comparison, we will also consider an empirically determined trapezoidal back-EMF waveform (obtained from [PPLH00]) with the same RMS value as the sinusoidal back-EMF waveform. Both back-EMF waveforms are shown in figure 5.

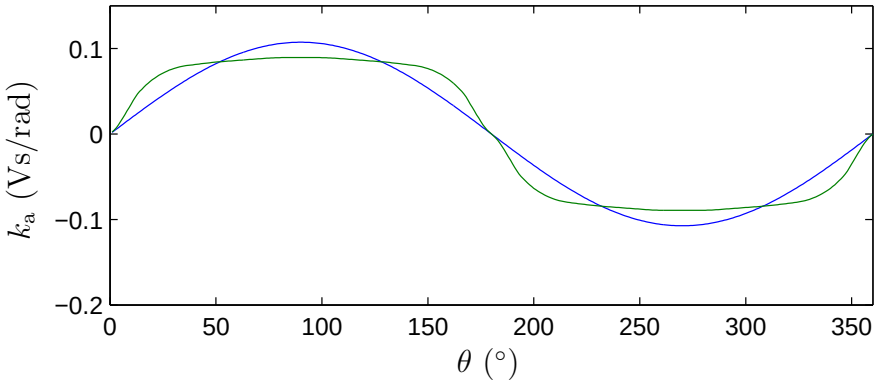

Figure 5: Sinusoidal and trapezoidal back-EMF waveform for the BLDC example.

6.1 Sinusoidal vs. rectangular back-EMF

We demonstrate the solution of (10) for both the sinusoidal and the trapezoidal back-EMF waveforms for $\tau^{\text{des}} = 0.3$ Nm. In figure 6, The optimal waveforms for $\omega = 300$ rad/s, which is below the rated speed of the motor, are shown in figure 6, We first note that the optimal current waveforms for the sinusoidal back-EMF are sinusoidal, verifying the classical result that sinusoidal currents simultaneously minimize power loss and acheive smooth torque.

The optimal waveforms for the trapezoidal motor are more subtle. Because the constraints are inactive, (10) can be solved in exactly one iteration of the operator splitting algorithm with ρ set to zero. Also, if eddy current is neglected, the optimal current waveforms are independent of ω. (This is not the case for delta connected motors, in which the back-EMF can induce circulating current which cannot be controlled by the interver.)

We also compare the optimal waveforms at $\omega = 425$ rad/s, which is above rated speed of the motor. In this case, the optimal current waveforms are no longer sinusoidal, for either back-EMF waveform.

6.2 Optimal vs. sinusoidal current waveforms

In the last example, we saw that sinusoidal current waveforms are not optimal in the constant power region for all values of λ, even with a sinusoidal back-EMF waveform. Here we show that, for $\omega = 425$ rad/s and $\tau^{\text{des}} = 0.3$ Nm, there is no value of λ for which sinusoidal current waveforms are optimal. Figure 7 shows the optimal waveforms for $\lambda = 0$ kW/(Nm)^2. $\lambda = 2$ kW/(Nm)^2, and as $\lambda \to \infty$. As expected, we find that larger values of λ result in lower torque ripple, but require greater phase advance and a greater amplitude current waveform. We also show the optimal sinusoidal currents for comparison, which also can be found using convex optimization, by solving a special case of the harmonic mitigation problem of §3.2. (Because sinusoidal current waveforms always produce a smooth torque output, the choice of λ is irrelevant.) Note that the sinusoidal current waveforms have noticably higher peak value than any of the other waveforms. When compared with the optimal waveform for

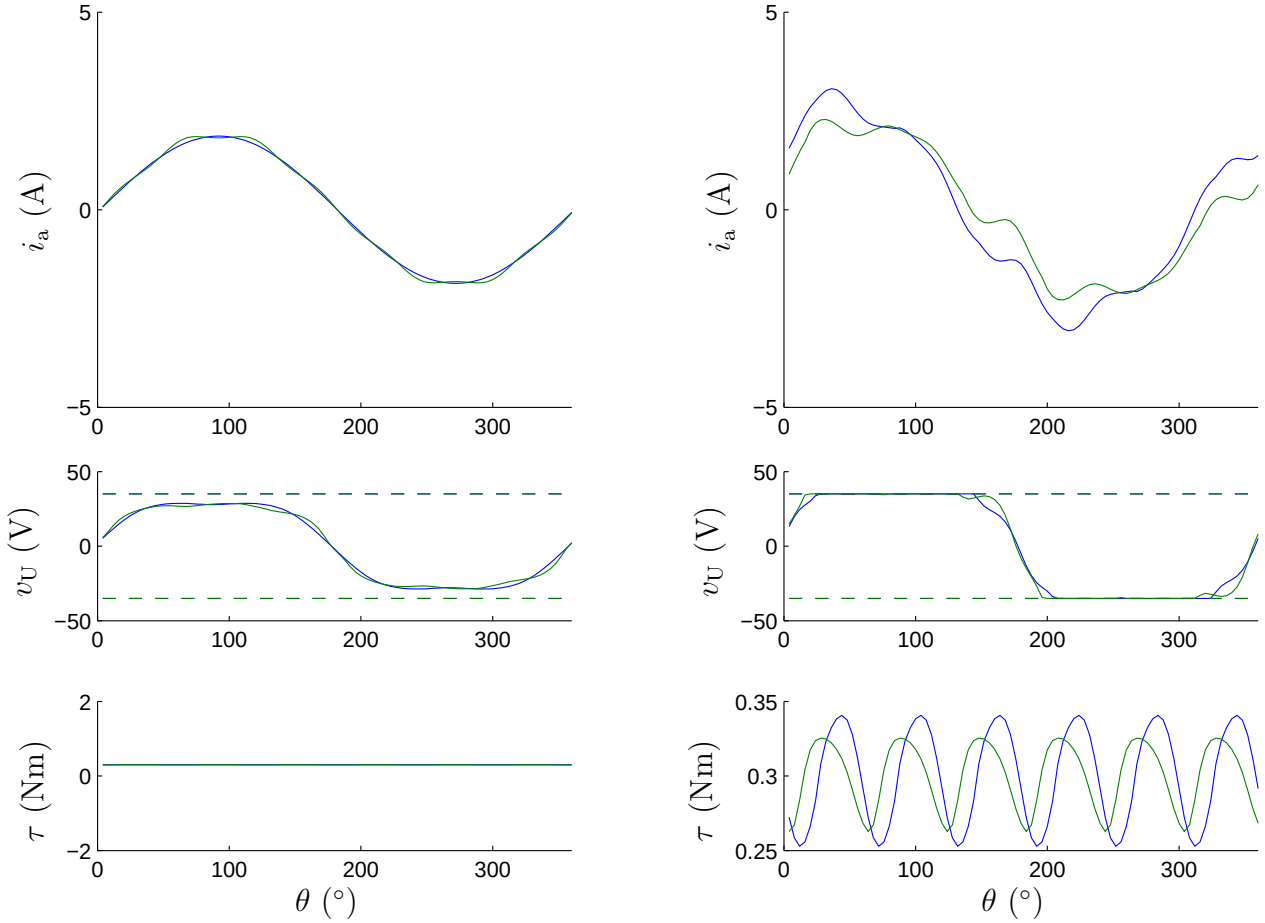

Figure 6: The optimal current and voltage waveforms for the both back-EMF waveforms, for $\tau^{\text{des}} = 0.3$ Nm and $\lambda = 2$ kW/(Nm)^2. The left and right figures show $\omega = 300$ rad/s and $\omega = 425$ rad/s, respectively, and the blue and green lines correspond to the sinusoidal and trapezoidal back-EMF waveforms, respectively.

$\lambda \to \infty$, the sinusoidal waveform has a greater magnitude for all θ, which immediately indicates higher power loss.

By varying λ, we can characterize the entire trade-off curve between power loss and torque ripple (shown in figure 8). Any point on this curve is optimal for some positive value of λ, and the points corresponding to the waveforms of figures 6 and 7 are shown. The point corresponding to the optimal sinusoidal current waveforms dose not lie on the curve, indicating that they are not optimal for any positive λ. Indeed, by using the optimal waveforms at this point, we can increase efficiency by several percent, depending on our tolerance for torque ripple.

We also show the curve corresponding to the optimal waveforms for the trapezoidal back-EMF, which in this case strictly outperforms the sinusoidal back-EMF, assuming optimal waveforms are used.

6.3 Open-phase fault

We demonstrate the ability of a delta connected motor to operate if one winding has failed in open circuit, and we find that this is possible, even with active voltage constraints. Note that in this case, only the first of the three symmetry assumptions (pole symmetry) holds. In figure 9, we show the optimal waveforms for $\omega = 650$ rad/s and $\tau^{\text{des}} = 0.3$ Nm), taking $\lambda \to \infty$, thus generating smooth torque.

7 Conclusion

In this paper, we pose torque control of brushless permanent magnet motors with PWM inverters as an optimal control problem. In this problem, we minimize a (user-defined) combination of power loss and torque ripple while achieving a desired average torque and respecting bridge voltage and phase current limits (arising from saturation). The resulting problem (or one of the proposed variations) is convex, and can therefore be solved quickly and reliably. We give an operator splitting algorithm which is fast enough to be implemented in real time (0.4 ms/solve), possibly on embedded platforms, and we give some practical recommendations to ensure a fast response (tens of microseconds) to changes in the desired torque.

We conclude by noting that for many classes of AC motors, (*e.g.*, induction machines and switched reluctance machines), a similar torque control problem would be a nonconvex optimization problem (due to nonlinear dynamics or a nonlinear relation between torque and current). Nonconvex optimization problems are hard to solve in general, and the operator splitting algorithm is not gauranteed to converge. We note, however, that for many nonconvex optimization problems, operator splitting methods appear to work well in practice (see [BPC+11] for details).

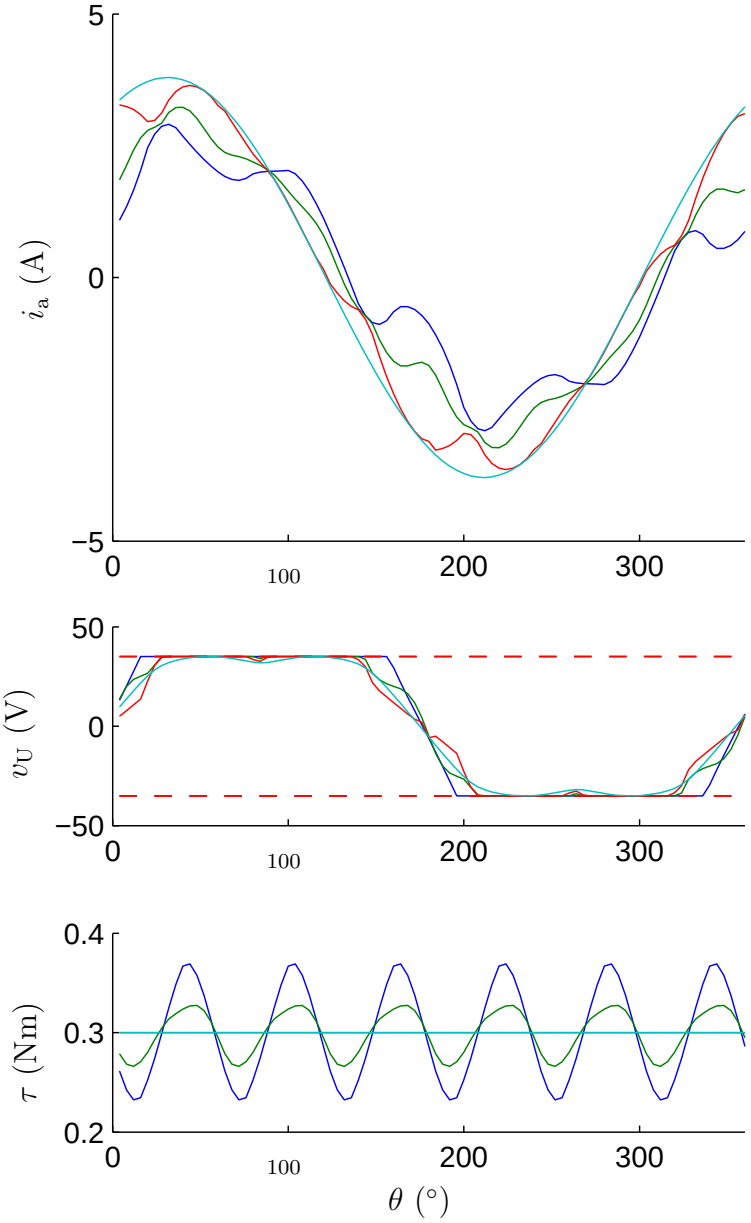

Figure 7: The optimal waveforms for the sinusoidal back-EMF, for $\omega = 425$ rad/s, and $\tau^{\mathrm{des}} = 0.3$ Nm, for three values of lambda, with the optimal sinusoidal currents shown for comparison. The blue, green, and red lines give correspond to $\lambda = 0$ kW/(Nm)^2, $\lambda = 2$ kW/(Nm)^2, and $\lambda \to \infty$, respectively, and the cyan line corresponds to the sinusoidal current.

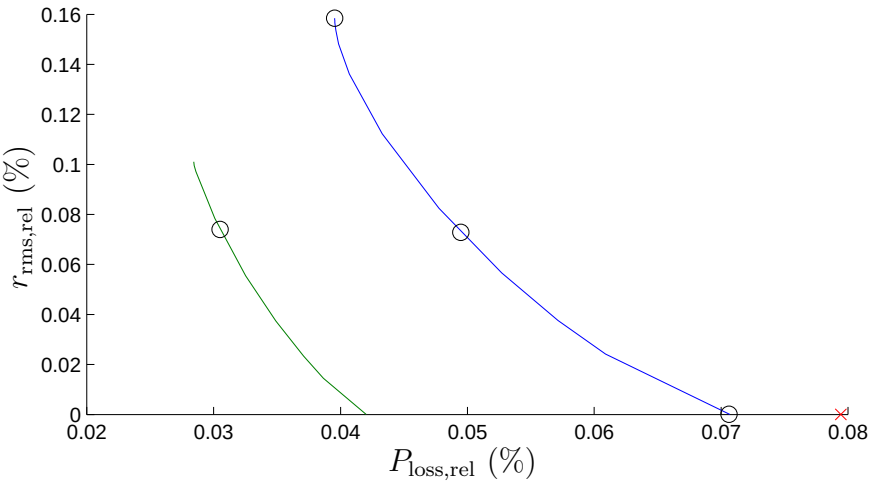

Figure 8: The achievable relative power loss and relative torque ripple for $\omega = 425$ rad/s and $\tau^{\mathrm{des}} = 0.3$ Nm, for both back-EMF waveforms (sinusoidal in blue and trapezoidal in green). The circles and triangle correspond to the waveforms shown in figures 6, and 7 respectively, and the $\times$ corresponds to the optimal sinusoidal waveforms.

References

[ABH01] F. Aghili, M. Buehler, and J. M. Hollerbach. Quadratic programming in control of brushless motors. In *Proc. of the 2001 IEEE International Conference on Robotics and Automation*, volume 2, pages 1130–1135, 2001.

[ABH03] F. Aghili, M. Buehler, and J. M. Hollerbach. Experimental characterization and quadratic programming-based control of brushless-motors. *IEEE Transactions on Control Systems Technology*, 11(1):139–146, 2003.

[BBPZ09] S. Bolognani, S. Bolognani, L. Peretti, and M. Zigliotto. Design and implementation of model predictive control for electrical motor drives. *IEEE Transactions on Industrial Electronics*, 56(6):1925–1936, 2009.

[BKKP11] S. Bolognani, R. Kennel, S. Kuehl, and G. Paccagnella. Speed and current model predictive control of an IPM synchronous motor drive. In *IEEE International Electric Machines & Drives Conference*, pages 1597–1602. IEEE, 2011.

[BMDL13] F. Baudart, E. Matagne, B. Dehez, and F. Labrique. Optimal current waveforms for torque control of permanent magnet synchronous machines with any number of phases in open circuit. *Mathematics and Computers in Simulation*, 90:1–14, 2013.

[BPC+11] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

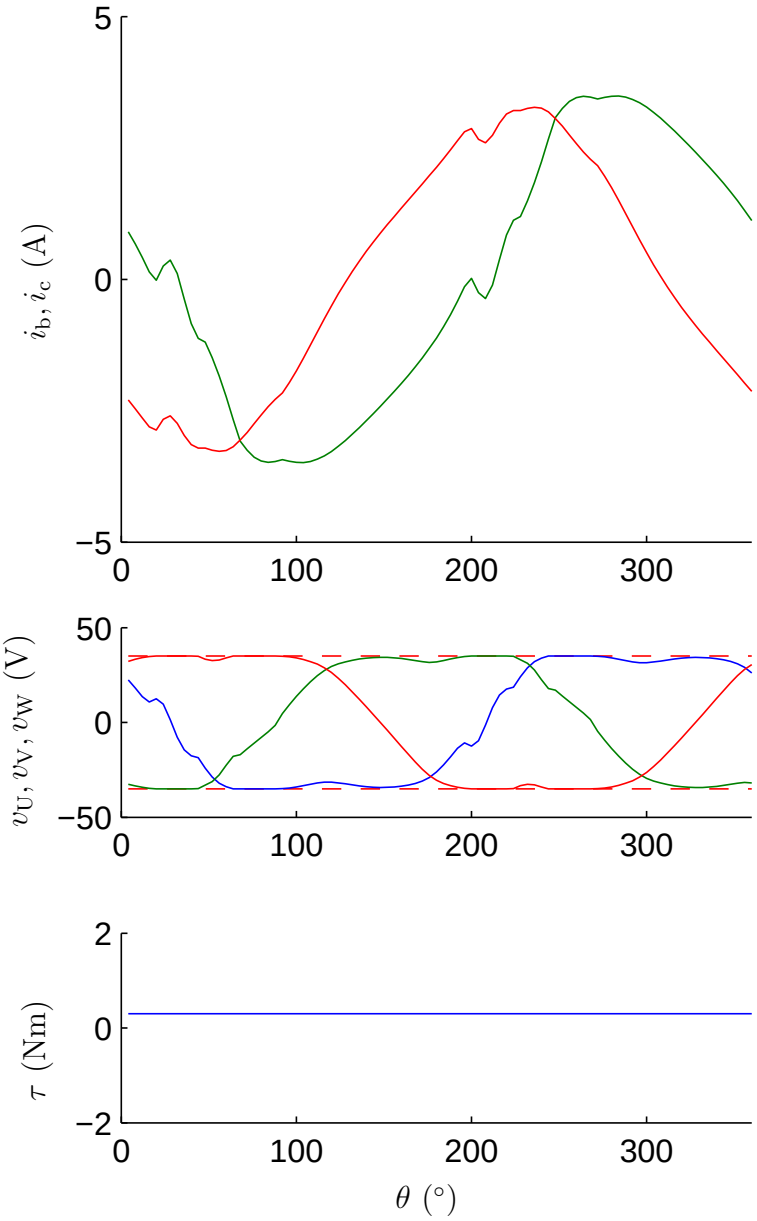

Figure 9: The minimum-ripple optimal current and voltage waveforms for $\omega = 650$ rad/s, $\tau^{\text{des}} = 0.3$ Nm), for a delta connected motor with a single open-phase fault.

[BV04] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[CCL08] K. T. Chau, C. C. Chan, and C. Liu. Overview of permanent-magnet brushless drives for electric and hybrid electric vehicles. *IEEE Transactions on Industrial Electronics*, 55(6):2246–2257, 2008.

[CN88] R. S. Colby and D. W. Novotny. An efficiency-optimizing permanent-magnet synchronous motor drive. *IEEE Transactions on Industry Applications*, 24(3):462–469, 1988.

[CPDB13] E. Chu, N. Parikh, A. Domahidi, and S. Boyd. Code generation for embedded second-order cone programming. In *Proc. of the 2013 European Control Conference*, pages 1547–1552, 2013.

[CSW99] P. L. Chapman, S. D. Sudhoff, and C. A. Whitcomb. Optimal current control strategies for surface-mounted permanent-magnet synchronous machine drives. *IEEE Transactions on Energy Conversion*, 14(4):1043–1050, 1999.

[Dav05] T. A. Davis. Algorithm 849: A concise sparse Cholesky factorization package. *ACM Transactions on Mathematical Software*, 31(4):587–591, 2005.

[GB08] M. Grant and S. Boyd. Graph implementations for nonsmooth convex programs. In V. Blondel, S. Boyd, and H. Kimura, editors, *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pages 95–110. Springer-Verlag Limited, 2008.

[GB13] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming. [Online]. Available: http://cvxr.com/cvx, September 2013.

[Gey11] T. Geyer. Computationally efficient model predictive direct torque control. *IEEE Transactions on Power Electronics*, 26(10):2804–2816, 2011.

[GLN80] R. Gabriel, W. Leonhard, and C. J. Nordby. Field-oriented control of a standard AC motor using microprocessors. *IEEE Transactions on Industry Applications*, 16(2):186–192, 1980.

[Han94] D. Hanselman. Minimum torque ripple, maximum efficiency excitation of brushless permanent magnet motors. *IEEE Transactions on Industrial Electronics*, 41(3):292–300, 1994.

[HD92] J. Y. Hung and Z. Ding. Minimization of torque ripple in permanent magnet motors: A closed form solution. In *Proc. of the 18th IEEE Industrial Electronics Conference*, pages 459–463. IEEE, 1992.

[HM94] J. R. Hendershot and T. Miller. *Design of brushless permanent-magnet machines*. Motor Design Books, 1994.

[JSHR06] Y. Jeong, S. Sul, S. Hiti, and K. M. Rahman. Online minimum-copper-loss control of an interior permanent-magnet synchronous machine for automotive applications. *IEEE Transactions on Industry Applications*, 42(5):1222–1229, 2006.

[LHPF86] H. Le-Huy, R. Perret, and R. Feuillet. Minimization of torque ripple in brushless DC motor drives. *IEEE Transactions on Industry Applications*, 22(4):748–755, 1986.

[LNCK07] J. Lee, K. Nam, S. Choi, and S. Kwon. A lookup table based loss minimizing control for FCEV permanent magnet synchronous motors. In *IEEE Vehicle Power and Propulsion Conference*, pages 175–179. IEEE, 2007.

[LZH05] Y. Liu, Z. Q. Zhu, and D. Howe. Direct torque control of brushless DC drives with reduced torque ripple. *IEEE Transactions on Industry Applications*, 41(2):599–608, 2005.

[MB10] J. Mattingley and S. Boyd. Automatic code generation for real-time convex optimization. In Y. Eldar and D. Palomar, editors, *Convex Optimization in Signal Processing and Communications*, pages 1–41. Cambridge University Press, 2010.

[MB12] J. Mattingley and S. Boyd. CVXGEN: a code generator for embedded convex optimization. *Optimization and Engineering*, 13(1):1–27, 2012.

[MDM09] S. Mariethoz, A. Domahidi, and M. Morari. Sensorless explicit model predictive control of permanent magnet synchronous motors. In *IEEE International Electric Machines and Drives Conference*, pages 1250–1257. IEEE, 2009.

[MTTH94] S. Morimoto, Y. Tong, Y. Takeda, and T. Hirasa. Loss minimization control of permanent magnet synchronous motor drives. *IEEE Transactions on Industrial Electronics*, 41(5):511–517, 1994.

[MWB11] J. Mattingley, Y. Wang, and S. Boyd. Receding horizon control: Automatic generation of high-speed solvers. *IEEE Control Systems Magazine*, 31(3):52–65, 2011.

[OSB13] B. O'Donoghue, G. Stathopoulos, and S. Boyd. A splitting method for optimal control. *IEEE Transactions on Control Systems Technology*, 21(6):2432–2442, Nov 2013.

[PB14] N. Parikh and S. Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):123–231, 2014.

[PPLH00] 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. *IEEE Transactions on Industrial Electronics*, 47(1):109–114, 2000.

[SDK12] J. Stumper, A. Dötlinger, and R. Kennel. Classical model predictive control of a permanent magnet synchronous motor. *European Power Electronics and Drives Journal*, 22(3):24–31, 2012.

[TTT99] K. Toh, M. J. Todd, and R. H. Tütüncü. SDPT3a MATLAB software package for semidefinite programming, version 1.3. *Optimization methods and software*, 11(1-4):545–581, 1999.

[TTT03] R. H. Tütüncü, K. C. Toh, and M. J. Todd. Solving semidefinite-quadratic-linear programs using SDPT3. *Mathematical programming*, 95(2):189–217, 2003.

[VJR99] S. Vaez, V. I. John, and M. A. Rahman. An on-line loss minimization controller for interior permanent magnet motor drives. *IEEE Transactions on Energy Conversion*, 14(4):1435–1440, 1999.

[WB10] Y. Wang and S. Boyd. Fast model predictive control using online optimization. *IEEE Transactions on Control Systems Technology*, 18(2):267–278, 2010.

[WC05] A. P. Wu and P. L. Chapman. Simple expressions for optimal current waveforms for permanent-magnet synchronous machine drives. *IEEE Transactions on Energy Conversion*, 20(1):151–157, 2005.

[WsCY+14] L. Wang, s. Chai, D. Yoo, L. Gan, and K. Ng. *PID and Predictive Control of Electrical Drives and Power Converters*. John Wiley and Sons, 2014.

[YWWL04] Y. Yang, J. Wang, S. Wu, and Y. Luh. Design and control of axial-flux brushless DC wheel motors for electric vehicles—Part II: Optimal current waveforms and performance test. *IEEE Transactions on Magnetics*, 40(4):1883–1891, 2004.